

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



TRABAJO FIN DE GRADO

Aplicaciones de Inteligencia Artificial en Nanotecnología

Andrés Ruiz Carrasco

Tutor: Sacha Gómez

Enero 2016

Resumen

En el campo de la adquisición de imágenes en Nanotecnología, donde los objetos a observar tienen un tamaño del orden de unos pocos nanómetros, el procesamiento por ordenador es la única forma viable de convertir las complejas señales de los microscopios en un formato de imagen intuitivo para los usuarios. Desde que se emplean microscopios capaces de observar objetos nanométricos, ha sido imprescindible el desarrollo de software a medida para el tratamiento de imágenes a esta escala. Debido a que las técnicas de microscopía atómica siguen evolucionando de forma constante y adquiriendo cada vez más resolución, es necesario un avance en paralelo de las aplicaciones de visualización de imágenes.

En este proyecto se ha realizado un estudio de imágenes tomadas por un microscopio electrónico de escala nanométrica, en concreto un microscopio de efecto túnel, se ha abordado de forma directa los principales problemas del software existente, como son las imágenes escalonadas, imágenes que poseen mediciones a distintas alturas, o imágenes con mucho ruido o poca resolución. Se han desarrollado dos aplicaciones completas, la primera, *AtomDetector*, es la encargada del análisis y detección de átomos en estas imágenes, mientras que, *AtomClassifier*, se ocupa de analizar cada uno de los átomos detectados, crear un modelo de cada uno de ellos y realizar una clasificación según su elemento.

Durante el proyecto se han implementado algoritmos propios tanto para el análisis de imágenes, delimitación de los átomos individuales y extracción de información de los mismos.

Para el proceso de clasificación se ha optado por implementar una red neuronal, en concreto un perceptron multicapa con backpropagation.

Además se ha desarrollado interfaces de relación con el usuario para facilitar el uso de estas aplicaciones.

Palabras Clave

Nanotecnología, átomo, microscopía, STM, detección, análisis de imágenes, machine learning, perceptrón.

Abstract

In the field of imaging in nanotechnology, where the observed objects have a size on the order of a few nanometers, the computer processing is the only viable way to convert the complex signals microscopes in a intuitive format image for the users. Since microscopes capable of observing nanoscale objects are used, it has been essential to develop custom software for image processing at this scale. Because atomic microscopy techniques are evolving steadily and becoming increasingly resolution, we need a parallel advance of image display applications.

This project has made a study of images taken by an electron microscope nanoscale, specifically a scanning tunneling microscope, has directly addressed the main problems of the existing software, such as staggered images, images that have measurements at different heights, or images with noisy or low resolution. We have developed two complete applications, the first, AtomDetector, is responsible for the analysis and detection of atoms in these images, while AtomClassifier, deals with analyzing each of the detected atoms, creating a model of each of them and a classification according to their element.

During the project own algorithms have been implemented for both image analysis, demarcation of individual atoms and extracting information from them.

For the classification process has been chosen to implement a neural network, in particular a multilayer perceptron with backpropagation.

Moreover, it has developed interfaces with the user relationship to facilitate the use of these applications.

Keywords

Nanotechnology, atom, STM, microscopy, detection, image analysis, machine learning, perceptron.

Agradecimientos

En primer lugar agradecer al grupo del profesor Miquel Salmeron en el Lawrence Berkeley National Laboratory, ya que cedió las imágenes sobre las que se han trabajado a lo largo de todo el proyecto.

También agradecer a los profesores de la universidad con los que he tenido el placer de coincidir en las clases, todos han aportado su grano de arena en mi formación, algunos con sus conocimientos y otros en el ámbito profesional con sus forma de trabajar, aprendiendo de las cosas que se deben hacer y otras que no. En especial a mis tutores durante el proyecto, ellos me han guiado durante el desarrollo del mismo consiguiendo al final un resultado espectacular desde mi punto de vista.

Por su puesto agradecer a mis padres el esfuerzo para darme la oportunidad de formarme en la universidad y poder realizar lo que de verdad me gustaba, y claro por su apoyo durante todos los años en los que he sido estudiante de la escuela, sin este no hubiese sido capaz de alcanzar mis metas.

No me puedo olvidar de mis amigos tanto los de la escuela como los de la infancia, siempre he tenido su apoyo y comprensión cuando no he podido verles durante días o semanas por tener que estudiar o hacer prácticas, siempre han estado ahí cuando les necesitaba.

Por último agradecer a Cristina, que es sin duda lo mejor que me llevo de todos estos años en la escuela, sin su apoyo y su ejemplo, trabajadora y constante como la que más, me hubiese sido imposible alcanzar mis metas.

Gracias a todos por el apoyo y esfuerzo depositado en mí.

Andrés Ruiz Carrasco

Enero 2016

Índice

1. Introducción	1
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.3 Estructura del trabajo.....	3
2. Estado del arte.....	5
2.1 La Nanotecnología.....	5
2.2 El átomo.....	5
2.3 Microscopía	7
2.3.1 Microscopía electrónica de barrido	7
2.4 Tratamiento y análisis de imágenes.....	8
2.5 Redes neuronales	10
3. Datos de partida	13
3.1 Contenido de los datos.....	13
3.2 Estructura de los datos	13
4. Análisis.....	15
4.1 Puntos críticos	15
4.2 Normalización	15
4.3 Análisis estadístico	16
4.4 Modelos de análisis	17
4.4.1 Modelo de planos paralelos.....	17
4.4.2 Modelo basado en media.....	18
4.4.3 Modelo de máximos y mínimos	19
4.4.4 Modelo de búsqueda por cuadrantes	19
4.4.5 Modelo final completo	20
4.5 Diseño e implementación	21
4.5.1 Diagrama UML	22
4.5.2 Diseño de clases e interfaces.....	24
4.5.3 Casos de uso	26
5. Modelado y clasificación	33
5.1 Aclimatación.....	33
5.2 Delimitación	34
5.3 Modelado	34

5.4	Clasificación	35
5.5	Diseño e implementación	36
5.5.1	Diagrama UML	36
5.5.2	Diseño de clases	38
5.5.3	Casos de uso	39
6.	Pruebas y resultados	47
6.1	Detección de puntos críticos en imágenes escalonadas	47
6.2	Entrenamiento y test sobre imágenes del mismo conjunto	49
6.3	Entrenamiento y test sobre imágenes sin restricciones	50
7.	Conclusiones y trabajo futuro	51
7.1	Conclusiones.....	51
7.2	Trabajo futuro	52
	Referencias	53

Índice de Figuras

Figura 1. Diagrama UML de la aplicación AtomDetector.	23
Figura 2. Diagrama de casos de uso AtomDetector	26
Figura 3. Ventana principal de la aplicación AtomDetector.	28
Figura 4. Ventana para la carga de una imagen de la aplicación AtomDetector.....	28
Figura 5. Representación de una imagen y estadísticas en la aplicación AtomDetector.	29
Figura 6. Análisis de una imagen sin tratamiento en la aplicación AtomDetector.	30
Figura 7. Análisis de una imagen con tratamiento en la aplicación AtomDetector.	31
Figura 8. Diagrama UML de la aplicación AtomClassifier.....	37
Figura 9: Diagrama de casos de uso AtomClassifier.....	40
Figura 10. Ventana principal de la aplicación AtomClassifier.....	41
Figura 11. Delimitación de una imagen mediante la aplicación AtomClassifier.	43
Figura 12. Clasificación de una imagen mediante la aplicación AtomClassifier.....	45
Figura 13. Detección de puntos críticos de una imagen escalonada utilizando MinMax.	48
Figura 14. Detección de puntos críticos de una imagen escalonada utilizando el modelo final.....	48
Figura 15. Entrenamiento y test de una imagen perteneciente al conjunto medido con H ₂ O adherido.	49
Figura 16. Entrenamiento y test de una imagen sin restricciones.	50

1. Introducción

1.1 Motivación

A lo largo de los últimos años, la elevada aparición de ordenadores, la infraestructura de Internet y la telefonía móvil, impulsaron el crecimiento de la industria de los semiconductores [1]. Hoy en día, estamos entrando en un mundo, en el cual los objetos a nuestro alrededor se convierten en microsistemas inteligentes que interactúan, a través de sensores y actuadores inalámbricos, con otros dispositivos y con las personas. Estamos inmersos dentro de un entorno inteligente.

Cuando hacemos referencia a ambientes inteligentes, debemos tener en cuenta un concepto importante, el de la Inteligencia Ambiental [2]. Este tipo de Inteligencia representa un entorno natural con tecnología oculta, a diferencia de la electrónica a la que actualmente estamos acostumbrados. Ahora bien, para ello, con el objetivo de llegar a todos los escenarios posibles, la interfaz hombre/máquina tiene que ser simple y la tecnología debe estar disponible en todo momento.

Por su parte, tanto en el área de la medicina como en el de los cuidados de la salud, se espera un gran cambio para las generaciones venideras, lo que trae aparejado la necesidad de cambios en las capacidades de computación y de redes disponibles hoy en día. La *Nanotecnología*, en cuanto a salud se refiere, es quien juega un rol fundamental en todo ello. Es la ciencia del futuro que nos lleva a la miniaturización, con alto grado de procesamiento de información. En [2] se afirma que la *Nanotecnología* cobra relevante importancia tanto en la administración de medicinas como en la realización de diagnósticos. Todo lo que se aplica en este ámbito de la medicina, puede ser llevado a cualquier otro dominio.

Sin embargo, para alcanzar esta tecnología, es necesario combinar diferentes áreas de trabajo con distintos especialistas, siendo así posible asegurar la construcción de sistemas complejos que puedan funcionar de acuerdo a las necesidades de aplicación.

Los recientes desarrollos en micro y Nanotecnología, el poder de cómputo actual y la comunicación inalámbrica proveen la plataforma ideal para realizar medidas biomédicas y, en general, la asistencia a los humanos. Es así como surge la *Nanociencia* [3], que se define como la ciencia que estudia las propiedades físicas, químicas y biológicas de los nanoobjetos (objetos en escala nanométrica), además de analizar cómo producirlos, y cómo pueden ser ensamblados por la autoorganización. Dentro de esta ciencia revolucionaria, surge la *Nanotecnología* [3], la cual cubre todos los métodos que pueden ser utilizados para trabajar a una escala molecular para reorganizar los componentes de los objetos y materiales, incluso progresando a la escala macroscópica.

Trabajar a escala nanométrica, independientemente del ámbito en que nos encontremos, tiene una serie de características intrínsecas que hace del uso de ordenadores algo imprescindible. Considerando que la luz visible tiene longitudes de onda de cientos de nanómetros, es evidente que todo tipo de microscopía óptica queda descartada en la toma de imágenes a escala nanométrica. Por tanto, cualquier método que busque la visualización del mundo a esa escala debe emplear una señal diferente a la interacción electromagnética en el rango visible. En todos los casos posibles, si se quiere hacer llegar al ser humano una imagen que sea capaz de analizar, hace falta un tratamiento previo de la señal que ha de ser realizado por ordenadores. La importancia del tratamiento de imágenes por ordenador en este campo se demuestra a través de los grandes esfuerzos que hacen las empresas de Nanotecnología en la contratación de informáticos y el desarrollo de software [4]. Es interesante mencionar el ejemplo de Nanotec Electrónica, que durante años ha seleccionado estudiantes de la Escuela Politécnica Superior (UAM) para el desarrollo de software de tratamiento de imágenes. En este escenario es en el que se desarrolla el presente trabajo, que busca diseñar nuevos algoritmos de tratamiento de imágenes de microscopía de campo cercano, donde se engloban una serie de técnicas de análisis de superficies muy extendidas, como pueden ser *Atomic Force Microscopy* (AFM) [5], *Electrostatic Force Microscopy* (EFM) [6] o *Scanning Tunneling Microscopy* (STM) [7].

1.2 Objetivos

En este trabajo se realizarán aplicaciones basadas en algoritmos diseñados a medida para llevar a cabo tareas de clasificación de muestras a escala nanométrica. Estos algoritmos se basarán en técnicas básicas de aprendizaje automático, que se modificarán para aumentar su eficiencia en tareas relacionadas con microscopía de fuerzas, de efecto túnel, o cualquier tipo de microscopía de campo cercano.

En el presente trabajo, hablaremos de diferentes técnicas diseñadas para la implementación de un software que permita el mejor análisis, caracterización, tratamiento y clasificación de los datos obtenidos de microscopios a escala nanométrica.

El principal objetivo de este trabajo es emplear técnicas de tratamiento de imágenes para mejorar el procesamiento automático de imágenes que actualmente se puede encontrar en el software de simulación comercial. Es importante decir que el software que actualmente se distribuye junto a los microscopios de fuerzas o de efecto túnel (que son los dos tipos de microscopía de campo cercano más extendidas) tiene una serie de importantes algoritmos de tratamiento de imágenes. Por tanto, en este trabajo se han intentado cubrir algunas de las lagunas que todavía se encuentran en este tipo de programas, como puede ser la identificación de átomos en superficies escalonadas.

El segundo objetivo, más ambicioso, es la identificación del tipo de átomos que se encuentran en la superficie. Como demostraremos a lo largo del trabajo, ésta es una tarea muy compleja y en la que incluso algoritmos mucho más sofisticados que los del primer punto no son capaces de realizar una clasificación correcta en los casos más difíciles. La dificultad de esta tarea es tal que, cuando ocurren avances significativos en la materia,

son publicados en revistas de altísimo impacto. Un ejemplo lo tenemos en la referencia [8], que fue portada de la revista Nature en 2007. En el presente trabajo se ha hecho un intento de clasificación de átomos a partir de las imágenes filtradas obtenidas tras el primer objetivo. Los resultados, aunque negativos en la mayor parte de los casos, se muestran y analizan en la segunda parte del trabajo.

1.3 Estructura del trabajo

La memoria de este trabajo consta de los siguientes apartados.

En el apartado 2 (Estado del arte) se explican los conocimientos necesarios para entender el resto del trabajo así como la situación actual de los temas que serán recurrentes a lo largo del documento: la definición exacta de Nanotecnología, aspectos relevantes de un átomo, los microscopios de sonda de barrido, el tratamiento y análisis de imágenes y las redes neuronales artificiales.

En el apartado 3 (Datos de partida) se lleva a cabo un estudio inicial de los datos de disponibles para la realización de este trabajo, esto es, los datos en bruto, sin ningún tratamiento, obtenidos de una colaboración entre el profesor Sacha Gómez de la EPS y el grupo del profesor Miquel Salmeron en el *Lawrence Berkeley National Laboratory*. Se explica en detalle qué son estos datos, su distribución y qué información proporcionan.

En el apartado 4 (Análisis) se exponen los métodos numéricos usados para el correcto análisis y detección de puntos críticos en las imágenes. Así como el diseño del software necesario para este propósito, además de los métodos utilizados para la representación de los mismos.

En apartado 5 (Modelado y clasificación) se exponen el diseño y los algoritmos implementados para la caracterización y modelado de los átomos, así como el proceso seguido para su clasificación mediante redes neuronales.

En el apartado 6 (Pruebas y resultados) se exponen las diferentes pruebas junto con los resultados obtenidos que incluyen la exploración y el análisis de las relaciones y la influencia de los distintos parámetros.

Por último, en el apartado 7 (Conclusiones y trabajo futuro) se detallan los problemas encontrados a lo largo de la implementación y las conclusiones adquiridas durante el desarrollo de este trabajo. También se introducen los posibles caminos a seguir en un trabajo futuro.

2. Estado del arte

En este apartado introducimos y contextualizamos una serie de conceptos teóricos necesarios para comprender la naturaleza del proyecto, así como los principios básicos de las técnicas empleadas para su desarrollo. En concreto, hablaremos de la *Nanotecnología* y cómo algunos autores o científicos la aplican a diferentes ámbitos. También, introduciremos el átomo y algunos de sus aspectos más relevantes para este trabajo, y además, explicaremos los métodos de medición de propiedades en escalas nanométricas (microscopía), el tratamiento y análisis de imágenes, así como los principios básicos de las redes neuronales empleadas.

2.1 La Nanotecnología

La Nanotecnología (deriva del griego *nanno* que significa *enano*) es el control de la materia a escalas de entre 1 y 100 nanómetros [9].

La Nanotecnología comprende el estudio, diseño, creación, síntesis, manipulación y aplicación de materiales, dispositivos y sistemas funcionales a través del control de la materia a nanoescala, además de la explotación de fenómenos y propiedades de esa materia [10].

Este término fue acuñado en 1974 por el japonés Taniguchi Norio y fue Richard Feynman (premio Nobel de Física en 1965) con su famosa conferencia titulada “*Hay mucho lugar al fondo*”¹, quien marcó un hito para el desarrollo de la Nanotecnología, haciendo ver la posibilidad de mover las cosas átomo por átomo. El gran despertar de la Nanotecnología comenzó a partir de los años 80, a partir del desarrollo de una amplia gama de microscopios de sonda de barrido, que logran imágenes a escala atómica.

Cuando se manipula la materia a escala tan minúscula, presenta fenómenos y propiedades totalmente nuevas. Por ello, los científicos utilizan la Nanotecnología con el objetivo de crear materiales, dispositivos y sistemas novedosos, poco costosos y con propiedades únicas.

2.2 El átomo

El concepto de átomo (proviene del latín *atomum*, y éste del griego *ἄτομον* cuyo significado es *sin porciones, indivisible*) como bloque básico e indivisible que compone la materia del universo, fue postulado por la escuela atomista en la Antigua Grecia.

Sin embargo, no fueron considerados seriamente por los científicos hasta el siglo XIX, cuando fueron introducidos para explicar ciertas leyes químicas. Con el desarrollo

¹ <http://www.zyvex.com/nanotech/feynman.html>

de la física nuclear en el siglo XX se comprobó que el átomo puede subdividirse en partículas más pequeñas.

Los átomos son objetos muy pequeños con masas igualmente minúsculas: su diámetro y masa son del orden de la billonésima parte de un metro y cuatrillonésima parte de un gramo. Este hecho provoca que solamente puedan ser observados mediante instrumentos especiales tales como un microscopio de efecto túnel (estos instrumentos se describen en detalle en el siguiente apartado). Más de un 99,94 % de la masa del átomo está concentrada en su núcleo, en general, repartida de manera aproximadamente equitativa entre protones y neutrones. El núcleo de un átomo puede ser inestable y sufrir una transmutación mediante desintegración radioactiva. Los electrones en la nube del átomo están repartidos en distintos niveles de energía u orbitales, y determinan las propiedades químicas del mismo. Las transiciones entre los distintos niveles dan lugar a la emisión o absorción de radiación electromagnética en forma de fotones, y son la base de la espectroscopia.

A pesar de que, tal y como se ha explicado, el átomo significa indivisible, en realidad está formado por varias partículas subatómicas. El átomo contiene protones, neutrones y electrones, con la excepción del hidrógeno, que no contiene neutrones, y del catión hidrógeno (o hidrón), que no contiene electrones. Los protones y neutrones del átomo se denominan nucleones, por formar parte del núcleo atómico.

Los protones y neutrones de un átomo se encuentran ligados en el núcleo atómico, la parte central del mismo. El volumen del núcleo es aproximadamente proporcional al número total de nucleones, el número másico A , lo cual es mucho menor que el tamaño del átomo, cuyo radio es del orden de 10^5 fm o 1 \AA (\AA). Los nucleones se mantienen unidos mediante la fuerza nuclear, que es mucho más intensa que la fuerza electromagnética a distancias cortas, lo cual permite vencer la repulsión eléctrica entre los protones.

Los átomos de un mismo elemento tienen el mismo número de protones, que se denomina número atómico y se representa por Z . Los átomos de un elemento dado pueden tener distinto número de neutrones: se dice entonces que son isótopos. Ambos números conjuntamente determinan el núclido.

Los átomos no están delimitados por una frontera clara, por lo que su tamaño se equipara con el de su nube electrónica. Sin embargo, tampoco puede establecerse una medida de esta, debido a las propiedades ondulatorias de los electrones. En la práctica, se define el radio atómico estimándolo en función de algún fenómeno físico, como la cantidad y densidad de átomos en un volumen dado, o la distancia entre dos núcleos en una molécula.

Los diversos métodos existentes arrojan valores para el radio atómico de entre $0,5$ y 5 \AA . Dentro de la tabla periódica de los elementos, el tamaño de los átomos tiende a disminuir a lo largo de un periodo, para aumentar súbitamente al comienzo de uno nuevo, a medida que los electrones ocupan niveles de energía más altos.

Las dimensiones del átomo son miles de veces más pequeñas que la longitud de onda de la luz ($400 - 700 \text{ nm}$) por lo que estos no pueden ser observados utilizando instrumentos ópticos. En comparación, el grosor de un cabello humano es equivalente a un millón de átomos de carbono, o si una manzana fuese del tamaño de la Tierra, los átomos en ella serían tan grandes como la manzana original.

El objetivo de este trabajo es, por tanto, a través de los datos recopilados por los dispositivos de medición especial, los cuales se detallan a continuación, diseñar un conjunto de algoritmos y un software capaz de diferenciar entre un tipo de átomos de otros de un elemento distinto, así como delimitar y caracterizar cada átomo individualmente.

2.3 Microscopía

Como se ha explicado anteriormente, el diminuto tamaño de los átomos impide que éstos sean medidos y observados con instrumentos de microscopía óptica, ya que la potencia amplificadora de un microscopio óptico está limitada por la longitud de onda de la luz visible.

Sin embargo, existen novedosos sistemas de medición que nos permiten medir estas partículas nanométricas, por ejemplo, las *microscopías electrónicas* durante este proyecto se han utilizado imágenes tomadas gracias a estos instrumentos. El microscopio electrónico utiliza electrones para iluminar un objeto y, dado que los electrones tienen una longitud de onda mucho menor que la de la luz, pueden mostrar estructuras mucho más pequeñas. La longitud de onda más corta de la luz visible es de alrededor de 4.000 \AA frente a la de los electrones que se utilizan en los microscopios electrónicos es de alrededor de $0,5 \text{ \AA}$.

Existen diversos sistemas de instrumentación, pero en este trabajo solamente se tratará con uno de los más básicos: el *microscopio electrónico de barrido*.

2.3.1 Microscopía electrónica de barrido

Una forma eficiente de obtener imágenes de la superficie de una muestra consiste en escanear la superficie con un haz de electrones en un patrón de rastreo, de forma parecida al barrido de un haz de electrones por la pantalla de una televisión. También se puede obtener información sobre superficies mediante una sonda de barrido en la que la trayectoria del haz de electrones atraviesa las regiones de particular interés. O como es el caso, el barrido se puede realizar con una sonda que monitoriza los electrones que atraviesan los túneles entre la superficie y la punta de la sonda, e incluso mediante una sonda que monitoriza la fuerza ejercida entre la superficie y la punta de la sonda.

En cualquier caso, el instrumento empleado para estas mediciones es un *microscopio electrónico de barrido* (*Scanning Electron Microscope*, SEM) [11] que crea una imagen ampliada de la superficie de un objeto. El SEM explora la superficie de la imagen punto por punto. Los electrones del haz pueden dispersarse de la muestra o provocar la aparición

de electrones secundarios. Los electrones perdidos y los secundarios son recogidos y contados por un dispositivo electrónico situado a los lados del espécimen. Cada punto leído de la muestra corresponde a un píxel en un monitor de televisión, además, cuanto mayor sea el número de electrones contados por el dispositivo, mayor será el brillo del píxel en la pantalla. A medida que el haz de electrones barre la muestra, se presenta toda la imagen en el monitor. Los microscopios electrónicos de barrido pueden ampliar los objetos 200.000 veces o más. Este tipo de microscopio es muy útil porque produce imágenes tridimensionales realistas de la superficie del objeto.

Las imágenes que se emplean en este trabajo están tomadas con un microscopio de efecto túnel (STM), una de las técnicas de microscopía de barrido más extendida, del grupo del profesor Miquel Salmeron en el *Lawrence Berkeley National Laboratory*. Son superficies de Rutenio en las que se han adsorbido moléculas de agua y carbono. La combinación de estas moléculas en la superficie ha generado distintas especies, como pueden ser CO, CH, H, O, etc.

2.4 Tratamiento y análisis de imágenes

El desarrollo de la microinformática en los últimos años ha propiciado que cada vez sea más común el trabajar con imágenes digitalizadas. La imagen digital se ha incorporado a la práctica en numerosos equipos de microscopía electrónica y microscopía confocal, además de sustituir a las tradicionales cámaras fotográficas en los microscopios ópticos. Las ventajas de la imagen digital frente a la analógica vienen dadas por las múltiples posibilidades de manipulación que nos ofrece. Además podemos procesar dichas imágenes para obtener diversa información. Reconocer, contar y medir tamaño, forma, posición o densidad de determinados objetos, como por ejemplo el área de los núcleos en una imagen histológica o, en este proyecto, la clasificación de distintos átomos, es algo que, con una correcta preparación de la muestra, está al alcance de cualquier ordenador personal complementado con un software adecuado de análisis de imágenes.

Desde un punto de vista físico, una imagen puede considerarse como un objeto plano cuya intensidad luminosa y color puede variar de un punto a otro. Si se trata de imágenes monocromas, esto es, en blanco y negro, se pueden representar como una función continua $f(x,y)$ donde (x,y) son sus coordenadas y el valor de f es proporcional a la intensidad luminosa (nivel de gris) en ese punto. Para obtener una imagen que pueda ser tratada por el ordenador es preciso someter la función $f(x,y)$ a un proceso de discretización tanto en las coordenadas como en la intensidad, a este proceso se le denomina digitalización.

La digitalización consiste en la descomposición de la imagen en una matriz de $M \times M$ puntos, donde cada uno tiene un valor proporcional a su nivel de gris. Dado que este valor puede ser cualquiera dentro de un rango continuo, es preciso dividir dicho rango en una serie de k intervalos, de forma que el nivel de gris de cada punto sea asignado a uno

de los valores que representa dicho intervalo. Los sistemas más modernos de proceso digital de imágenes suelen trabajar con 256 niveles de gris.

Una imagen se divide en una serie de elementos y cada uno de ellos recibe el nombre de píxel, *picture element*. El número de niveles de gris y las dimensiones de la matriz nos condicionan la capacidad de resolución de una imagen digital.

En el caso de imágenes en color, la intensidad puede considerarse como un vector tridimensional cuyas componentes son las intensidades en las tres bandas espectrales: rojo, verde y azul. Se habla entonces de una imagen multibanda, puesto que la información de la imagen color se desglosa en tres imágenes correspondientes a cada una de las bandas del espectro visible. Para cada una de dichas bandas se utilizan 8 bits de información, teniendo así un total de 24 bits, los cuales permiten obtener más de 16 millones de posibles combinaciones de colores. Además, en cuanto a las dimensiones de las imágenes digitales, éstas varían en función de las capacidades del sistema empleado. En general, se utiliza un octeto para almacenar cada píxel, por lo que las intensidades de las imágenes se cuantifican en 256 niveles (8 bits). Así, según estos tamaños, una imagen digital con 256 niveles de gris y una dimensión de 256×256 píxeles, ocupará 64 KB de memoria, la misma imagen con una dimensión de 512×512 píxeles ocupará 256 KB y si se trata de una imagen de 1024×1024 llegará a necesitar 1 MB de memoria. Además, si se tiene esta misma imagen en color, con tres bandas, se necesitarían 3 MB de memoria para ser almacenada.

Puesto que el objetivo de este trabajo es mejorar el procesamiento automático de imágenes actual e identificar en ella diferentes tipos de átomos es necesario cuantificar automáticamente los determinados objetos presentes en cada imagen y, para ello, el ordenador debe realizar una clasificación. Esta clasificación consiste en un proceso en el que los píxeles pertenecientes a una imagen se dividen en clases, normalmente dos: objetos de interés y fondo. Los objetos de interés pueden ser núcleos en imágenes histológicas, áreas de fibrosis en tejido hepático, vainas de mielina en axones nerviosos o, en este caso, átomos.

Un método sencillo de clasificación es el denominado “*thresholding*”, que se utiliza para convertir una imagen de niveles de gris en una imagen binaria. Cada píxel es clasificado como *ON* blanco, con un valor de 255, u *OFF* negro, valor 0, dependiendo de si su nivel de gris excede o no un valor umbral. La selección de este umbral se realiza a partir de un estudio del histograma de niveles de gris de la imagen. Este método es el utilizado durante el presente trabajo para presentar al usuario un primer vistazo de la imagen a analizar.

La clasificación puede realizarse de modo interactivo visualizando cómo queda la imagen según varía el valor del umbral, o de modo automático, donde el ordenador determina el valor de los umbrales de segmentación siguiendo diversos criterios [12] [13]. En el caso de imágenes en color la detección se puede realizar en función del histograma de intensidades de cada una de las bandas, roja, verde y azul, que componen la imagen.

La imagen binaria resultante del proceso de clasificación puede que no represente perfectamente los objetos de interés debido a problemas inherentes en la imagen original, como objetos que se tocan, o problemas del propio proceso de las imágenes, por ejemplo, pequeñas áreas mal clasificadas. Con el fin de solucionar estos problemas se pueden aplicar distintos operadores morfológicos para separar zonas, cribar áreas de determinados tamaños, erosionar o dilatar regiones, extraer bordes, entre otros.

Dos de las operaciones morfológicas más utilizadas en el análisis de imágenes son la erosión y la dilatación, las cuales se detallan a continuación.

La primera de ellas, consiste en examinar cada píxel y cambiarlo de *ON* a *OFF* si alguno de sus vecinos está en *OFF*. Normalmente se utilizan como vecinos los 8 que rodean al píxel examinado, aunque para algunas aplicaciones se pueden utilizar conectividades de 4 vecinos, los 2 verticales y los 2 horizontales e, incluso, conectividades de 2 vecinos, los verticales o los horizontales. Como se explicará más adelante, en este proyecto se han tenido en cuenta 8 vecinos, ya que se consideran importantes los píxeles, en este caso puntos, en diagonal.

La segunda operación, la dilatación, es el proceso inverso, que consiste en cambiar los píxeles de *OFF* a *ON* si alguno de sus vecinos esta *ON*. Al resultado de una erosión más una dilatación se le denomina apertura, “*opening*”. El nombre proviene de la tendencia de esta secuencia de operaciones a separar, abrir, puentes de unión entre objetos próximos o a abrir cavidades próximas al borde. La operación opuesta, esto es, dilatación más erosión, es denominada cierre, “*closing*”, y puede usarse para conectar objetos muy próximos o para rellenar pequeños huecos.

2.5 Redes neuronales

Las redes de neuronales artificiales (RNA) son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas o redes neuronales.

Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones: función de propagación, activación y transferencia.

1. La *función de propagación* que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión. Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria.
2. La *función de activación*, que modifica a la anterior. Puede no existir, siendo en este caso la salida la misma función de propagación.
3. La *función de transferencia*, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas.

Para el desarrollo de este proyecto se ha optado por diseñar una red neuronal basada en el perceptrón multicapa con retroalimentación (*backpropagation*) que sea capaz de realizar una clasificación de diferentes átomos.

3. Datos de partida

Como ya se ha explicado anteriormente en la sección 2.3 del estado del arte, se parte de un conjunto de datos recopilados mediante un sistema de microscopía del orden de los nanómetros, con el objetivo de desarrollar métodos capaces de diferenciar átomos de diferentes elementos, así como delimitar y caracterizar cada átomo de manera individual. De manera que, en este capítulo se detalla el contenido de las diferentes mediciones (datos) disponibles así como la estructura de las mismas.

3.1 Contenido de los datos

El contenido de los datos disponibles para llevar a cabo este trabajo son diferentes muestras tomadas por un sistema de microscopía de efecto túnel, técnica más extendida de microscopía de barrido, de una superficie de Rutenio sobre la que están en suspensión varios átomos de carbono (C), hidrogeno (H), oxígeno (O) y una molécula de agua (H_2O).

A partir de estos datos y como se detallará en las secciones posteriores, se diseñará una interfaz de fácil uso para el usuario que realice las tareas de representación, análisis y clasificación de los mismos.

3.2 Estructura de los datos

El conjunto de datos utilizado en este proyecto se compone de una serie de mediciones, o lo que a partir de ahora también denominaremos imágenes, separadas en diferentes ficheros siguiendo la siguiente estructura.

En primer lugar, se compone de una cabecera que muestra información relevante de la imagen, esto es, la escala, la cual es útil para obtener una perspectiva real de las dimensiones, junto con la relación de los ejes de la matriz y la amplitud de los mismos, las cuales proporcionan información acerca de la resolución real de estas mediciones. Toda esta información es esencial y necesaria para la correcta visualización e interpretación de las imágenes.

Además de esto, los ficheros contienen una matriz de tamaño $N \times M$, con datos que proporcionan información sobre la posición de la punta del microscopio que realiza la medición, como un punto (x, y) de la porción medida. Estos datos se utilizarán para la caracterización de los átomos así como para su representación gráfica.

Respecto a estos datos de la matriz, cabe destacar que, fijando uno de los ejes, ya sea x o y , se pueden obtener enfoques diferentes de los datos. Por ejemplo, si se fija el eje x de una matriz $N \times M$ se obtiene, para cada $x \in N$, una curva que representa el perfil de la imagen en ese punto, es decir, por cada $x \in N \rightarrow M$ puntos (y, z) .

Siguiendo este enfoque, desde el eje x se obtienen N perfiles y, por otra parte, M perfiles desde el eje y , pudiendo tratar estos perfiles como funciones continuas y así obtener máximos y mínimos tanto locales como globales de las mismas o realizar diversos cálculos sobre ellas.

Para realizar un posterior estudio dividido en zonas, ya que como se ha explicado en la sección 2.2, la medición de un átomo concreto influye en la medición de los átomos que se encuentran a su alrededor, se divide la matriz en lo que este trabajo se ha denominado cuadrante. Estos cuadrantes se tratan de unas sub-matrices o porciones de la matriz original que, en algunos de los algoritmos que más tarde se expondrán, se tratarán como matrices de datos independientes. Cabe destacar, y se tomará como norma general en el desarrollo de este proyecto, que la matriz de datos se separa en porciones de igual tamaño intentado realizar una división para obtener un total de 48 sub-matrices independientes.

4. Análisis

Una vez identificados los datos de partida, se puede proseguir con la siguiente fase del proyecto, realizando un análisis de estos datos para la posterior detección de puntos críticos en la matriz de datos.

De este modo, en este apartado se detallan las propiedades de los denominados puntos críticos, así como los métodos de normalización, el análisis estadístico realizado sobre los datos iniciales y los diferentes modelos de análisis empleados. Además, se incluyen algunos detalles de la implementación llevada a cabo para este análisis.

4.1 Puntos críticos

A lo largo de este trabajo, se considera como punto crítico aquellos puntos que tienen gran probabilidad de ser parte de un átomo concreto. Esta probabilidad será calculada mediante una serie de algoritmos que se aplicarán a la matriz de puntos.

Para llevar a cabo la identificación de estos puntos críticos se han utilizado una serie de algoritmos, algunos son los utilizados por software especializado en el campo, como por ejemplo el corte por planos paralelos o el software WsXM [4], y otros son de desarrollo propio. En los apartados siguientes se profundizará en el funcionamiento de cada uno de estos algoritmos y se explicará el sistema de probabilidades usado para la detección de estos puntos.

4.2 Normalización

Antes de proceder con el análisis, se realiza la normalización de los datos, de esta manera se consigue que los cálculos sean menos costosos y se simplifiquen.

En primer lugar, se genera una copia de la matriz de datos a la que se va a aplicar la normalización. Esta normalización consiste en lo siguiente: para cada punto que contiene esa matriz, se suma el valor mínimo de la misma y la unidad, obteniendo así una matriz de datos con valores entre $[1, valorMinimo + 1]$. Tal y como se ha explicado, dicha normalización aligera y simplifica los cálculos necesarios para la identificación de átomos, ya que se reduce a operaciones con números positivos.

Es importante destacar que esta normalización solamente se tendrá en cuenta a la hora de la identificar de manera superficial los posibles puntos críticos, y no se usarán estos datos normalizados cuando se caracterice cada átomo individualmente, puesto que el signo del dato es bastante significativo en estos casos.

Así, se obtiene lo que a partir de ahora se denominará matriz normalizada, la cual se denota como sigue.

$$Mnorm[x][y] = M[x][y] + valorMinimo + 1$$

A continuación, en el siguiente apartado, se procederá a estudiar diferentes tipos de algoritmos y enfoques con el objetivo de detectar con la máxima precisión posible este conjunto de puntos críticos, que más tarde se utilizarán para realizar la detección individual de cada átomo y su clasificación.

4.3 Análisis estadístico

Como punto de partida para la detección de puntos críticos se realiza un análisis estadístico de la matriz de datos, empezando primero con un estudio de la matriz al completo, más adelante para cada uno de los perfiles, mencionados en la sección 3.2 y, por último, este mismo estudio para cada cuadrante de la matriz. Cabe destacar que estos cálculos estadísticos se realizarán tanto para la matriz de datos original como para la matriz de datos normalizada, explicada en el apartado anterior.

Para la detección de los puntos críticos se realizan diferentes cálculos, descritos a continuación.

Desde el punto de vista global, teniendo en cuenta la matriz completa, se determina la media y mediana de los puntos, así como el punto máximo y mínimo. Además, para cada perfil x e y , se obtienen estos mismos datos estadísticos, junto con el crecimiento máximo, decrecimiento máximo, velocidad de crecimiento y velocidad de decrecimiento.

Finalmente, tratando cada uno de los cuadrantes como una matriz independiente, se calcularán los valores estadísticos de cada uno de ellos, a partir de las siguientes fórmulas.

Para obtener la media aritmética:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + a_3 + \dots + a_n}{n}$$

Siendo n el número total de puntos y cada a_i los datos de cada matriz.

Y para la mediana:

$$M_e = \begin{cases} a_{(n+1)/2} & \text{si } n \text{ es impar} \\ \frac{a_{n/2} + a_{(n/2)+1}}{2} & \text{si } n \text{ es par} \end{cases}$$

En el caso de los perfiles anteriormente citados, además de estos datos estadísticos, se han calculado lo que se ha denominado crecimiento máximo y decrecimiento máximo. Estos datos aportan información muy valiosa acerca de cómo se comporta cada átomo de manera individual ante la medición, a la vez que permiten determinar la velocidad con la que se alcanzan estos máximos y mínimos. Para la obtención de estos valores, se debería tratar cada perfil como una función continua, sin embargo, no ha sido posible interpolar los puntos individuales para poder utilizar métodos convencionales de cálculo diferencial, ya que estos métodos degradarían información valiosa para una clasificación posterior.

Por este motivo se han desarrollado métodos discretos para el cálculo tanto del crecimiento y decrecimiento máximo y velocidades de crecimiento y decrecimiento.

Como se ha explicado en la sección 3.2, las mediciones contienen información importante acerca de las relaciones y resoluciones de los ejes, lo que permite calcular este crecimiento y decrecimiento máximo y su velocidad.

De este modo, es posible obtener el crecimiento máximo partiendo del punto máximo y buscando, en todos los puntos que conforman el perfil, aquel en el que los valores empiezan a crecer en dirección hasta dicho punto máximo; la diferencia entre este punto y el máximo es el buscado crecimiento máximo. De forma análoga se puede obtener el decrecimiento máximo.

Una vez se conoce el número de puntos que tarda en alcanzar este punto máximo o mínimo y, gracias a la información aportada en las mediciones, se puede obtener lo que denominamos velocidad de crecimiento o decrecimiento.

Adicionalmente, cada uno de los cuadrantes en los que se ha dividido la matriz de datos, pueden ser analizados de forma global, teniendo en cuenta todos los puntos contenidos en cada uno de ellos, o por perfiles. Basándonos en estos cuadrantes también se han calculado los mismos datos estadísticos que para la matriz completa, pero tratando cada uno de ellos como una matriz de datos independiente.

4.4 Modelos de análisis

Para la interpretación de los datos obtenidos durante el análisis estadístico y diferencial discreto, se han desarrollado distintos modelos de interpretación, los cuales presentan una serie de ventajas y desventajas. En este apartado se explicarán cada uno de estos métodos desarrollados, realizando además una comparativa entre ellos, para así seleccionar el más adecuado que se utilizará en la segunda fase del proyecto, es decir, para la clasificación y caracterización de los átomos.

En este punto solamente se realiza un análisis teórico de cada uno de ellos, y no se muestran ejemplos de las pruebas realizadas, ya que éstas se presentarán posteriormente en una sección dedicada a ellas.

4.4.1 Modelo de planos paralelos

La primera opción que se ha llevado a cabo se trata de una de las técnicas usadas por el software actual encargado de realizar análisis similar a los expuestos en este documento. Este modelo es el conocido como planos paralelos.

Para la detección de puntos críticos, este modelo selecciona dos planos paralelos al plano xy , los cuales tienen una altura calculada mediante el uso de los valores máximos y mínimos de la matriz de datos global junto con una tolerancia ajustable entre $[0, 1]$. De esta forma se tiene la matriz de datos M con sus puntos máximo P_{max} y mínimo P_{min} y dos planos paralelos T y F a una altura h_{top} y h_{floor} definidas por las ecuaciones:

$$\begin{cases} h_{top} = (tolerancia \cdot P_{max}) \\ h_{floor} = (tolerancia \cdot P_{min}) \end{cases}$$

Quedando los planos definidos por:

$$\begin{cases} z = h_{top} \\ z = h_{floor} \end{cases}$$

Este modelo indica que todos los puntos de la matriz M que queden por encima del plano T ($z = h_{top}$) o por debajo del plano F ($z = h_{floor}$) son considerados puntos críticos.

Se trata de un modelo sencillo que es útil para generar una primera aproximación, aunque totalmente inútil para un análisis en detalle de imágenes complejas, aquellas que contengan escalones o conjuntos de átomos a diferentes alturas. En el software desarrollado en este proyecto este modelo se identifica como modelo básico.

4.4.2 Modelo basado en media

La segunda opción de análisis que se ha desarrollado consiste en un método nuevo que se basa en cómo se comportan cada uno de los perfiles de la matriz. Una vez obtenidos los datos estadísticos de cada uno de los perfiles, tanto los correspondientes al eje x como al eje y , se utiliza como solución a los problemas del modelo anterior, ya que al analizar cada perfil por separado permite una mejor definición de cómo se comporta cada átomo a lo largo de las regiones de la matriz.

Este segundo método, llamado *MeanBased* en el software desarrollado, se basa en el valor de la media de cada uno de los perfiles de la matriz, obteniendo así una mejor separación entre las distintas zonas de la matriz y evitando comportamientos anómalos.

A pesar de que se trata de un algoritmo más elaborado que la primera opción explicada, no resuelve los problemas con imágenes complejas, escalonadas o que contengan átomos en distintos niveles de altura. A lo que se añade otro problema adicional, ya que los perfiles pueden tener información sobre el comportamiento de zonas específicas, pero mediante este punto de vista se pierde la visión global de la matriz, tendiendo a clasificar puntos críticos cuando en realidad no lo son.

Para esta clasificación, se recorren los perfiles de la matriz junto con el valor medio de cada uno de ellos y una tolerancia entre $[0, 1]$ y se etiquetan como puntos críticos aquellos puntos cuyo valor distan mucho del valor calculado. Teniendo la frontera de clasificación siguiente:

$$\text{punto crítico si } \begin{cases} z < (\bar{x} - (tolerancia \cdot \bar{x})) \\ z > (\bar{x} + (tolerancia \cdot \bar{x})) \end{cases}$$

Como se verá más adelante este método por sí solo no soluciona los problemas que el primer método (modelo de planos paralelos) presentaba, pero con algunos ajustes y en combinación con otros métodos resultará muy útil a la hora de definir un algoritmo completo de análisis.

4.4.3 Modelo de máximos y mínimos

En tercer lugar se elabora un modelo basado en el mismo principio de analizar los perfiles de la imagen de forma independiente, pero en vez de tomar como referencia el valor de la media de los mismos (tal y como ocurría en el modelo anterior) se realiza una búsqueda de máximos y mínimos locales en dichos perfiles. En el desarrollo del software implicado en esta fase del proyecto se ha denominado *MinMax*.

Además, tal y como se ha explicado en la sección 4.3 del presente documento, se ha optado por un análisis discreto de los perfiles, evitando así la pérdida de información relevante al aplicar un método de interpolación. Por ello, para realizar una búsqueda de máximos y mínimos válida en este modelo, se ha implementado un algoritmo que recorre los perfiles buscando cambios en la pendiente de la curva, tomando como referencia conjuntos de varios puntos consecutivos.

El método considera como puntos críticos el conjunto de puntos en los que se detecte un cambio en el sentido de la pendiente.

A pesar de las favorables expectativas de este modelo, no se han conseguido resultados fiables. El motivo principal de ello es el ruido, debido a pequeñas variaciones en el valor de la altura de la punta del sistema microscópico, causado por la propia medición de los átomos, teniendo así varios conjuntos de puntos con cambios de pendiente casi inapreciables que son tomados como puntos críticos erróneamente.

Este algoritmo será mejorado en modelos más elaborados explicados en esta misma sección, además, será una parte principal en la selección de puntos críticos en el modelo de detección final. Cabe destacar que una de las ventajas de este método es que no utiliza una tolerancia ajustable como ocurría en los modelos anteriores, ya que proporciona una medición estándar sin necesidad de la intervención humana.

4.4.4 Modelo de búsqueda por cuadrantes

Tal y como se ha explicado anteriormente, la matriz de datos se puede dividir en cuadrantes independientes obteniendo así multitud de sub-matrices que son tomadas como matrices únicas de datos. Este modelo se basa en este proceso, realizando cálculos sobre cada matriz de manera independiente.

Para la implementación de este modelo, se parte de la premisa de que cada matriz es independiente y que cada una de ellas no influye en el resto, y es por ello que se toman cada una de estas matrices realizando un análisis con el modelo de planos paralelos y el modelo de máximos y mínimos. La unión de estos dos modelos implica que, para que en el modelo final, un punto sea considerado punto crítico éste debe estar incluido tanto en el conjunto resultante del primer modelo como en el siguiente. Mediante esta unión se consigue purgar el resultado final de puntos, los cuales por motivos ya explicados se clasificaban como puntos críticos cuando en realidad no lo eran.

En este caso, se han obtenido resultados bastante satisfactorios, aunque queda sin resolver el problema en el que los átomos se encuentran en la separación de dos matrices contiguas, problema con el que lidiaremos hasta el modelo final.

Este modelo es denominado *SectorSearch* en el software desarrollado, el cual será explicado en detalle, su diseño, implementación e interfaz más adelante en el apartado 4.5.

4.4.5 Modelo final completo

Tras diseñar diversos métodos, los cuales han sido explicados a lo largo del apartado 4, se han ido detectando varios fallos y el motivo de ellos en cada caso. A pesar de esto, se ha desarrollado un método final que une las ventajas de cada uno y minimiza sus debilidades.

Por tratarse de un método que minimiza los fallos, este método será el utilizado en la segunda fase del proyecto, que se centra en la delimitación de átomos independientes y su posterior modelado y clasificación.

Este modelo final se basa tanto en los perfiles de la matriz global como en los perfiles de cada uno de los sectores, teniendo en cuenta los distintos parámetros anteriormente detallados. Además, para poder dar una explicación detallada de cómo funciona este algoritmo se explican de forma descendente cada uno de los pasos seguidos, empezando por los datos obtenidos de la matriz global y terminando en los perfiles de las sub-matrices generadas.

Como principio básico de este modelo se establece lo que se denomina ganancia. La ganancia será el crecimiento neto de un punto específico a lo largo de un perfil de la matriz de datos, esto es, partiendo de un punto inicial P la ganancia respecto al punto siguiente será la diferencia de éstos, teniendo así una ganancia positiva si la curva crece, o negativa si la curva decrece.

Como punto de referencia a esta ganancia adquirida se toma lo que se denomina ganancia máxima, es decir, la diferencia entre el punto máximo de cada perfil respecto al punto mínimo del mismo.

$$G_{max} = P_{max} - P_{min}$$

Además de este punto de referencia, este método incluye una tolerancia, que como se comprenderá más adelante, solo proporciona una regulación de lo estricto que se quiera ser con la detección, de manera que se pueda variar este valor dependiendo del tipo de imagen que se estudie. Este hecho generaliza el algoritmo permitiendo utilizarlo en distintos tipos de mediciones y con resoluciones muy distintas.

Gracias al mencionado concepto de ganancia, se puede establecer un control sobre los problemas que presentaban otros modelos frente a errores de medición, ya que una pequeña variación en la pendiente del perfil no afectará al resultado final de la detección.

A continuación se procede a los detalles de los perfiles de las sub-matrices generadas. Partiendo de una matriz de datos como la detallada en la sección 3, en primer lugar, se

realiza da subdivisión en matrices más pequeñas, se estudian los perfiles de cada una de ellas y de este modo se obtiene un primer conjunto de puntos críticos.

En concreto, para realizar este análisis, se parte de la base de la ganancia, se recorre cada uno de los perfiles punto por punto buscando aquellos en los que la ganancia acumulada supere un rango el valor producido por la ecuación siguiente:

$$\text{punto critico si } \begin{cases} z < (-G_{\max} - (\text{tolerancia} \cdot -G_{\max})) \\ z > (G_{\max} + (\text{tolerancia} \cdot G_{\max})) \end{cases}$$

Además, se realizan dos búsquedas independientes, una para cada perfil, es decir, una para los perfiles x y otra para los perfiles y , de manera que para ser considerado punto crítico éste debe estar incluido en ambas búsquedas.

Como añadido y para contrastar este método, se analizan cada uno de los puntos críticos obtenidos por el presente método mediante el modelo de planos paralelos, donde aquellos puntos incluidos en ambas búsquedas serán directamente incluidos en la lista de puntos críticos final, mientras que aquellos que queden excluidos por el método de planos paralelos se comprobarán más adelante. Cabe destacar que los puntos que el método de planos paralelos categorice como punto crítico pero que por otra parte el método de la ganancia no lo haga, serán descartados automáticamente.

Por último, una vez estudiadas las sub-matrices, se realiza el estudio de la matriz global, primero, teniendo en cuenta aquellos puntos a los que el sistema de planos paralelos del paso anterior hubiese descartado y el de ganancia hubiese aceptado. En este caso, para el estudio de la matriz global se sigue el mismo procedimiento utilizado para las sub-matrices o cuadrantes. Esto es, en primer lugar, se recorren los perfiles comprobando si la ganancia adquirida sobrepasa cierto umbral, si un punto está incluido tanto en el método de los cuadrantes como en el de la matriz global, se considera punto crítico sin importar lo que haya predicho el método de planos paralelos. Por el contrario, si un punto es catalogado como punto crítico por el método de planos paralelos pero la búsqueda de ganancia, tanto a nivel global o como individual no ha resultado positivo se descarta como punto crítico.

Más adelante en el apartado 6 del presente documento se muestran ejemplos de selección de cada uno de los métodos implementados

4.5 Diseño e implementación

En los apartados anteriores se ha definido el proceso a seguir para la detección de puntos críticos en una matriz de datos así como el análisis estadístico. En este apartado se describen los detalles del diseño e implementación de la aplicación desarrollada, tales como el diagrama UML, la funcionalidad de cada una de las clases así como las relaciones entre las mismas o la interfaz de usuario, mostrando los casos de uso básicos de la aplicación.

En primer lugar, cabe destacar que se ha seleccionado el lenguaje de programación Java sobre su versión 1.7. Además, gracias a su característica máquina virtual ésta es multiplataforma, pudiendo ser utilizada en la mayoría de dispositivos, que junto con un

sistema de representación intuitivo y cómodo facilita al usuario las herramientas necesarias para este tipo de análisis.

Durante el desarrollo de este proyecto se ha seguido un ciclo de desarrollo de basado en un desarrollo iterativo e incremental, esta elección ha sido muy acertada ya que permite realizar y completar hitos concretos con resultados finales, pudiendo así desarrollar un sistema adaptable, dinámico y funcional desde el inicio del mismo.

De esta forma se generaban prototipos funcionales, cada poco tiempo, los cuales aportan una nueva funcionalidad o interfaz en cada iteración.

4.5.1 Diagrama UML

El modelo UML que siguen las clases más características del software implementado se muestra en la Figura 1. La aplicación desarrollada, *AtomDetector*, se ha realizado de manera modular con el objetivo de que, en un futuro trabajo, se puedan introducir nuevos métodos de análisis o representaciones de manera sencilla.

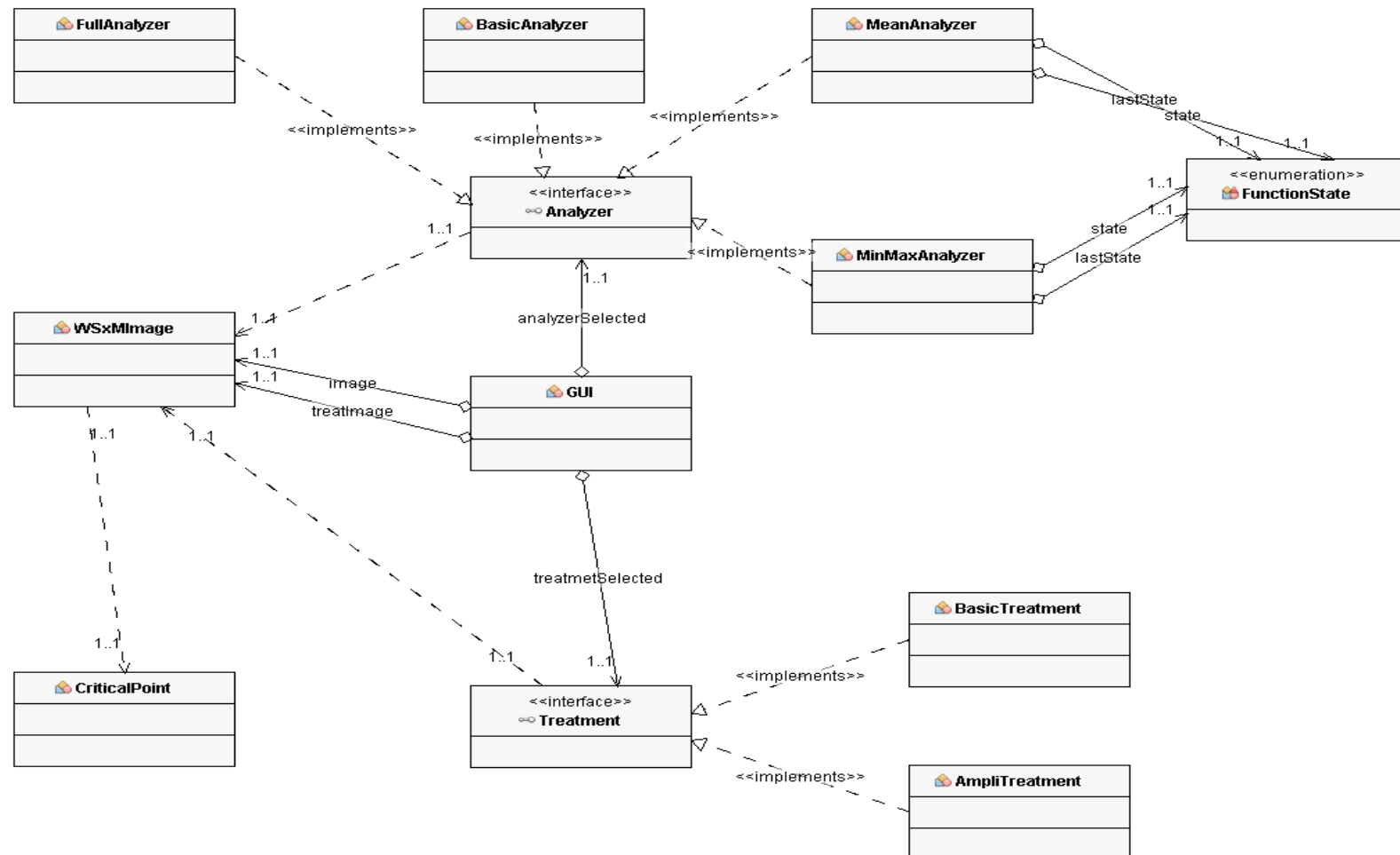


Figura 1. Diagrama UML de la aplicación AtomDetector.

4.5.2 Diseño de clases e interfaces

Tal y como se puede observar en el modelo UML de la sección anterior, la aplicación está dividida en varias clases principales. En esta sección se detallan los métodos más importantes así como su función en el conjunto del software implementado. Se comenzará explicando las clases que modelan los objetos tangibles, como son *WSxMImage*, *CriticalPoint* y la clase principal de la interfaz de usuario. A continuación, se documentarán las interfaces *Analyzer* y *Treatment* que corresponden a los modelos de análisis del apartado 4.4.

La aplicación está dividida en cinco paquetes bien definidos, cada uno de los cuales se describen a continuación.

- El primero de ellos, *data*, que contiene las clases encargadas del manejo de datos, como son *WSxMImage*, *CriticalPoint* y *FunctionState*.
- El paquete *analyzer* que contiene la interfaz del analizador así como las clases que la implementan. En este paquete se encuentran los métodos encargados del análisis de la matriz de datos y de la detección de los puntos críticos.
- En el paquete *treatment* se encuentra la interfaz de tratamiento, *Treatment*, y las clases que lo implementan. Dicha interfaz se encarga de realizar ciertas normalizaciones o modificaciones sobre la matriz de datos.
- Por último, en el paquete *ui*, se encuentra la clase encargada de implementar la interfaz gráfica que utilizará el usuario.

A continuación se expone una breve descripción de las clases implementadas de cada uno de los paquetes anteriores, así como su función en el conjunto de la aplicación.

4.5.2.1 WSxMImage

Esta clase es la encargada de modelar las imágenes tomadas por el software de medición *WSxMImage*. Para instanciar un objeto de esta clase es necesario proporcionar la ruta a un fichero que cumpla el formato especificado en la sección 3.

Una vez leída la matriz de datos y sus escalas, se realiza una normalización documentada en el apartado 4.2 de este mismo documento.

Además de modelar este tipo de imágenes, la clase *WSxMImage* posee métodos para su representación gráfica, mediante un mapa de calor, de la matriz de datos para así poder obtener una referencia visual de la imagen leída.

También, al mismo tiempo que se obtienen los datos, esta clase almacena los distintos perfiles y los datos estadísticos de la matriz de datos, como son el número de puntos, máximos, mínimos, media o mediana.

4.5.2.2 CriticalPoint

Esta clase es la encargada de modelar un punto, el cual se compone de unas coordenadas (x, y) , un valor z para indicar su altura y una serie de métodos que permiten realizar comparaciones.

4.5.2.3 FunctionState

FunctionState se trata de una enumeración que contiene los tres estados posibles de una función, esto es, creciente, decreciente o constante. Ésta será utilizada por ciertos modelos de análisis.

4.5.2.4 Analyzer

Es una interfaz que implementan las clases de analizadores, tal y como se han detallado en el apartado 4.4 del análisis, contiene las funciones y algoritmos necesarios para la detección de puntos críticos en la matriz de datos obtenida a partir de las imágenes.

Existen cinco clases que implementan esta interfaz, las cuales son: *BasicAnalyzer*, *MeanAnalyzer*, *MinMaxAnalyzer*, *SectorAnalyzer* y *FullAnalyzer*. Estas clases contienen cada uno de los tipos de analizadores implementados.

Cabe destacar que el método que otorga esta interfaz es *analyze*, el cual a partir de una matriz de datos proporciona una lista de puntos críticos (*CriticalPoint*).

4.5.2.5 Treatment

Se trata de una interfaz que implementa los tratamientos previos que se pueden aplicar sobre la matriz de datos, los cuales generan una nueva matriz con una serie de normalizaciones o transformaciones.

Existen tres clases de tratamientos que implementan esta interfaz: *NormalizeTreatment*, *AmpliTreatment* y *BasicTreatment*.

Sin embargo, el uso de estos métodos es opcional y dependiendo del tipo de imagen a analizar es conveniente utilizar uno u otro.

4.5.2.6 GUI

La interfaz gráfica de la aplicación se implementa en la clase *Gui*, que se encuentra en el paquete *ui*. Ésta es la clase ejecutable que genera la interfaz de usuario y otorga una visión de cada uno de los análisis posibles y las imágenes cargadas.

En la siguiente sección (apartado 4.5.3), se explican en detalle los casos de uso básicos para el correcto funcionamiento de la misma.

4.5.3 Casos de uso

Las clases y funcionalidades descritas en el apartado han sido integradas en una interfaz gráfica de fácil uso para el usuario, orientada al análisis y representación de los datos. A continuación, se presentarán los principales casos de uso de la aplicación desarrollada, incluyendo diversos ejemplos de la funcionalidad aportada por la misma.

De manera que se detallan los tres principales escenarios de uso de la aplicación *AtomDetector*, empezando por la carga de una imagen desde la interfaz de usuario (apartado 4.5.3.1), y finalizando por el análisis de una imagen cargada aplicando un tratamiento específico (apartado 4.5.3.3) estos escenarios pueden observarse en la Figura 2.

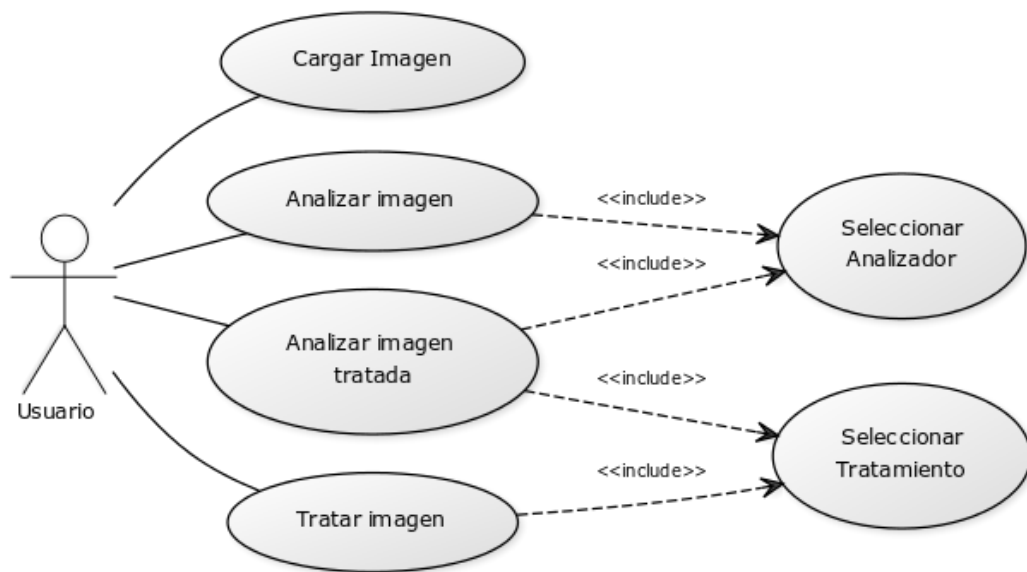


Figura 2. Diagrama de casos de uso AtomDetector.

Antes de continuar con la descripción de dichos casos de uso, se muestra en la Figura 3 la ventana principal de la aplicación, donde se ofrecen diferentes opciones y funciones que el usuario puede llevar a cabo. A continuación, se describen cada uno de los puntos que tiene esta interfaz con el usuario los cuales aparecen marcados en la Figura 3.

Open File (1)

Se encarga de abrir un archivo y se utiliza para seleccionar la imagen que va a ser analizada.

Clear (2)

Este botón reinicia los todos los campos de la interfaz, imágenes, análisis y estadísticas.

Analyzer (3)

Se trata de una lista desplegable donde el usuario puede seleccionar el analizador que desea utilizar.

Treatment (4)

Al igual que antes, es una lista desplegable en la que el usuario, de manera opcional, indica al sistema el tratamiento que desea aplicar a la matriz de datos y generar una nueva matriz con una serie de normalizaciones o transformaciones. Por defecto, si el usuario no selecciona nada, no se aplica ningún cambio en la matriz.

Original Image (5)

Es una pequeña ventana dentro de la interfaz donde se muestra la representación gráfica de la imagen a analizar. Un ejemplo de esto se incluye más adelante los apartados 4.5.3.2 y 4.5.3.3.

Treat Image (6)

En esta parte de la interfaz el usuario podrá visualizar la imagen junto con los puntos críticos obtenidos tras el análisis.

Statistics (7)

Se trata de un cuadro de texto donde se mostrarán los datos estadísticos de la imagen cargada.

Log (8)

En el campo *Log* se muestran cada una de las acciones que la aplicación ejecuta, de esta forma el usuario puede tener un historial de las diferentes actividades que se han llevado a cabo sobre la imagen.

Analyze (9)

Con este botón se inicia el proceso de análisis. En caso de que el usuario haya seleccionado un tratamiento, primero aplicará éste y a continuación el análisis.

Treat (10)

Este botón inicia el tratamiento seleccionado sobre la imagen.

Exit (11)

Para salir de la aplicación, existe el botón *Exit*, el cual libera todos los recursos utilizados y la cierra.

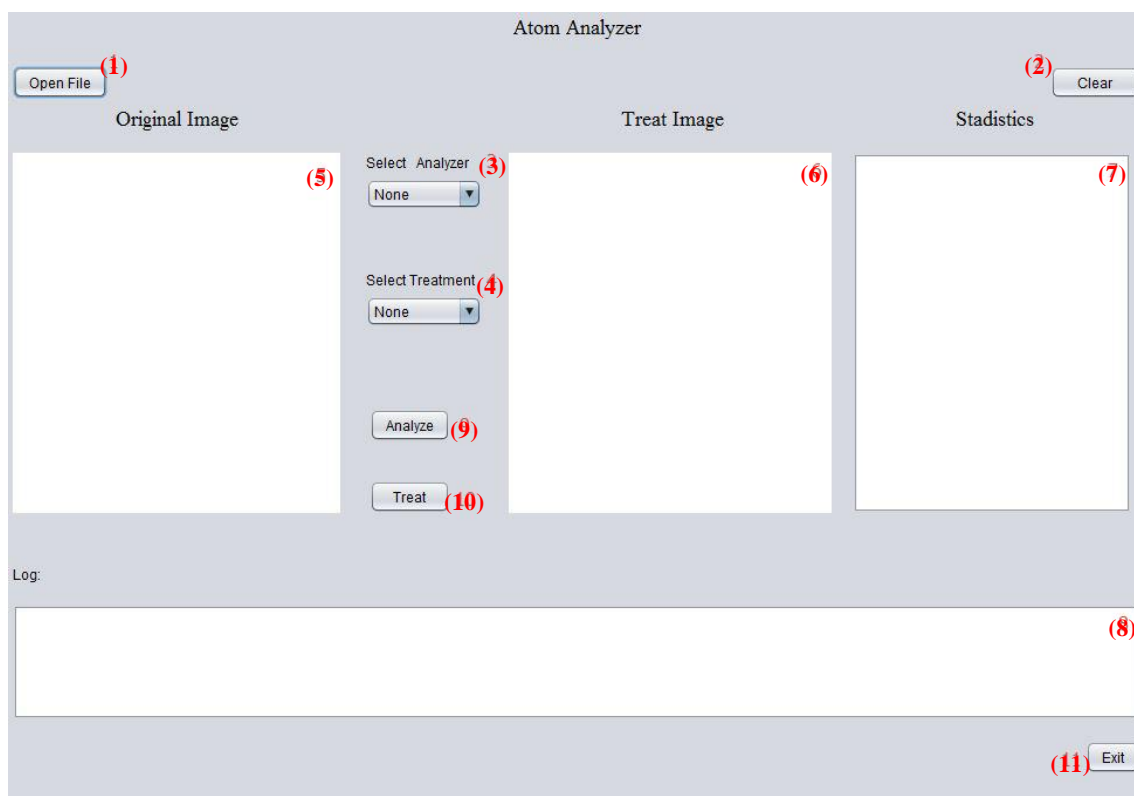


Figura 3. Ventana principal de la aplicación AtomDetector.

4.5.3.1 Carga de una imagen

En primer lugar, para llevar a cabo el análisis, es necesario que el usuario cargue una imagen. Para ello, basta con seleccionar el botón *Open File* (punto (1) en la Figura 3) y, a continuación, se abrirá una nueva ventana en la que el usuario puede seleccionar el archivo que desea analizar, tal y como se muestra en la Figura 4.

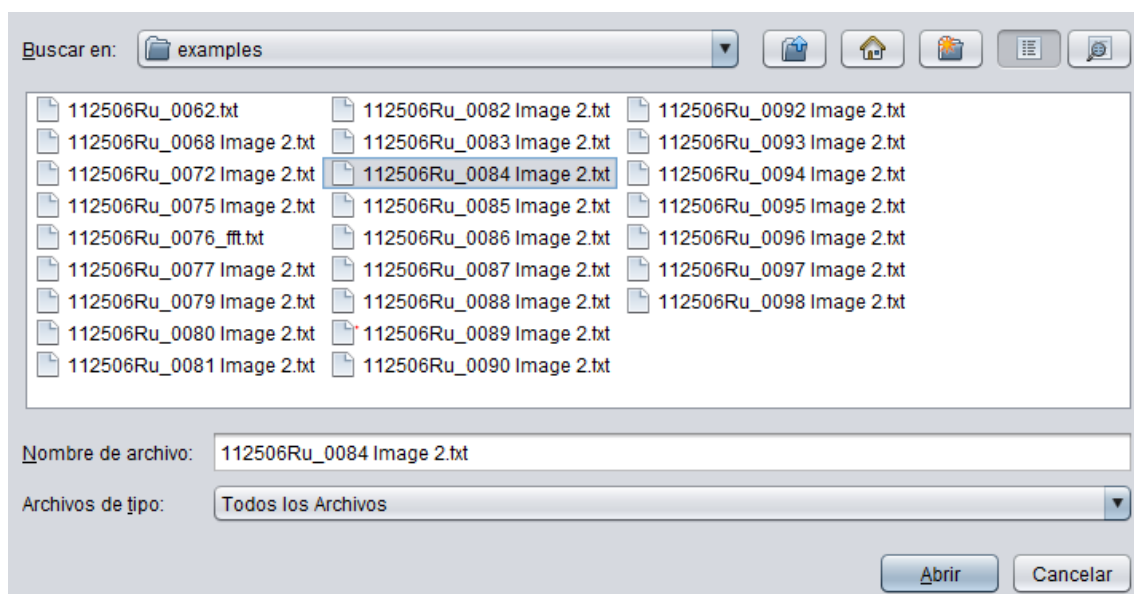


Figura 4. Ventana para la carga de una imagen de la aplicación AtomDetector.

Una vez seleccionado el archivo deseado, el programa cargará la imagen y mostrará los datos estadísticos de la misma, junto con su representación gráfica (véase Figura 5).

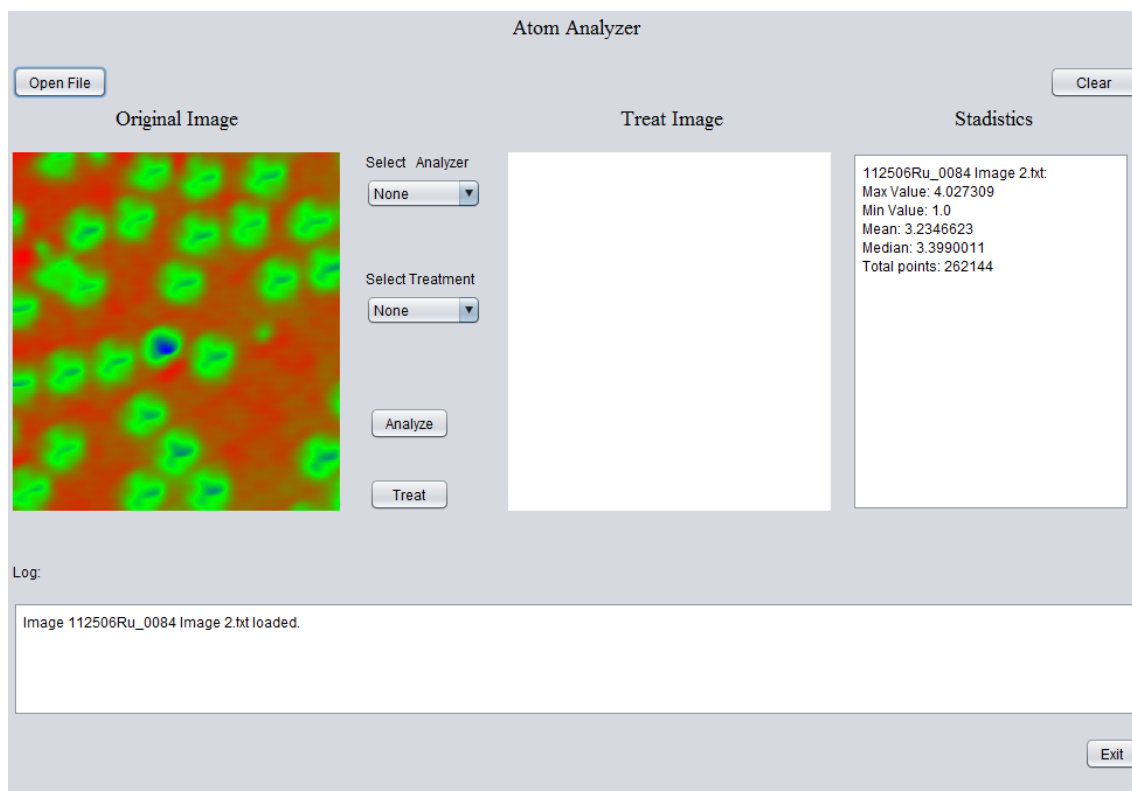


Figura 5. Representación de una imagen y estadísticas en la aplicación AtomDetector.

4.5.3.2 Análisis de una imagen sin tratamiento

Una vez cargada una imagen en la aplicación, el siguiente paso para su análisis es pulsar el botón *Analyze* (punto (9) en la Figura 3). En ese momento, puesto que se va a realizar un análisis sin tratamiento, la aplicación comienza el análisis y, tras unos minutos, mostrará en el apartado correspondiente (*Treat Image*) la imagen original junto con los puntos críticos detectados representados en un amarillo intenso. Esto se muestra en la Figura 6.

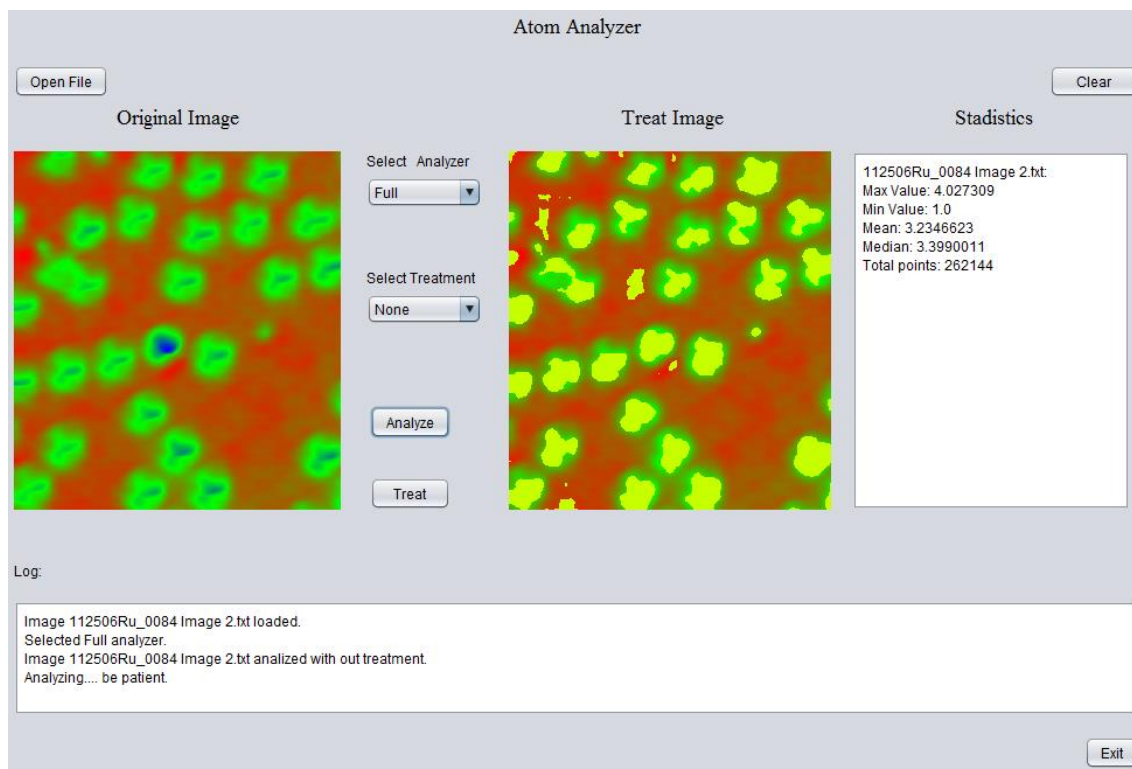


Figura 6. Análisis de una imagen sin tratamiento en la aplicación AtomDetector.

Una vez analizada la imagen el usuario podrá limpiar los datos mediante el botón *Clear* y seleccionar otra imagen para su análisis. Por otra parte para abandonar la aplicación bastara con presionar el botón *Exit*.

4.5.3.3 Análisis de una imagen con tratamiento

En el caso en el que se desee analizar una imagen que ha sido tratada anteriormente basta con pulsar el botón *Analyze* (punto (9) en la Figura 3) siendo necesario haber seleccionado el tratamiento a aplicar. La aplicación funciona de manera similar al caso anterior, mostrando la imagen con los puntos críticos. En la Figura 7 se muestra la salida del análisis con un tratamiento amplificador. Además de esto, es posible probar distintos tratamientos mediante el botón *Treat* (punto (10) en la Figura 3).

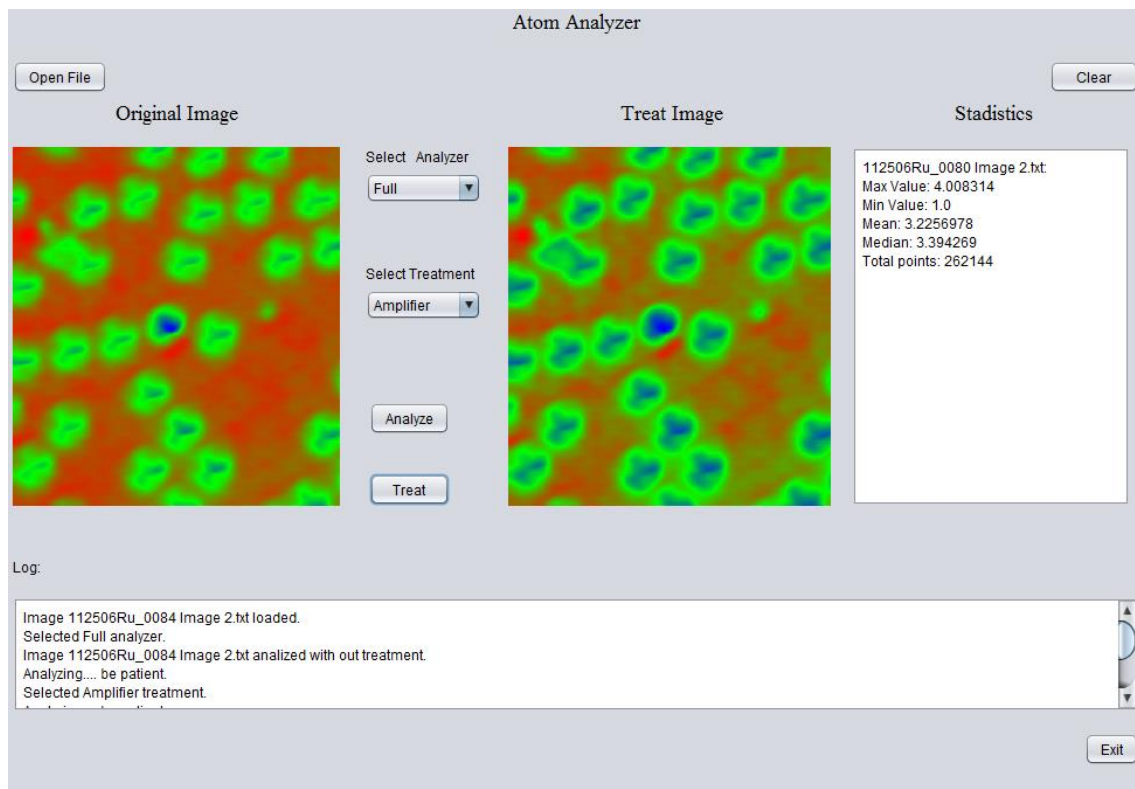


Figura 7. Análisis de una imagen con tratamiento en la aplicación AtomDetector.

5. Modelado y clasificación

Una vez analizadas las imágenes y detectados los puntos críticos de cada una de ellas explicado en el apartado anterior, para alcanzar el objetivo de este proyecto es necesario realizar el modelado de cada uno de los átomos de manera individual y así poder llevar a cabo la clasificación de los mismos.

Los diferentes pasos seguidos para esta modelización, así como las pertinentes normalizaciones y aclimataciones de la nube de puntos críticos, la cual fue obtenida mediante los métodos documentados anteriormente, se describen en esta sección.

Para ello, partimos del modelo final de análisis, ya que se trata del método más completo encontrado y mediante el cual se han obtenido los mejores resultados. De esta forma, utilizando la nube de puntos conseguida con dicho modelo, se ha diseñado una segunda aplicación que aclimata este conjunto de puntos para que sean útiles para una posterior delimitación, modelado y clasificación de los átomos individualmente de cada imagen.

5.1 Aclimatación

El primer paso, por tanto, para el modelado y clasificación de átomos es la aclimatación de los puntos críticos obtenidos mediante el análisis de la matriz de datos contenida en las imágenes. Para llevar a cabo esta aclimatación de la nube de puntos se ha seguido el siguiente procedimiento.

En primer lugar, se recorre cada punto de la imagen y para cada uno de ellos se comprueban los puntos que se encuentran a su alrededor. Al tratarse de una matriz de dos dimensiones, cada uno de estos puntos posee 8 vecinos, de forma que si al menos 6 de estos vecinos pertenecen a la nube de puntos críticos obtenida mediante el análisis previo (apartado 4), este punto se denotará como punto crítico por cercanía. Además de esta denominación, se le otorgará lo que en este proyecto hemos denominado índice de vecindad, esto es, el número de puntos críticos de la nube original con los que está en contacto. Por lo que estos nuevos puntos críticos tendrán un valor de este índice comprendido entre 6 y 8. Cabe destacar que, como se verá en apartados posteriores, es importante que estos puntos críticos, denotados así por cercanía, no tengan el mismo peso que los puntos críticos obtenidos durante el análisis original.

Este método sigue un esquema muy simplificado, ya que en este caso no existen tantas variables como en los diferentes campos en los que se suele utilizar. Está basado en los algoritmos de corrección de píxeles perdidos o defectuosos tomando como referencia valores correctos de los píxeles a su alrededor. Este tipo de algoritmos son muy utilizados en campos como la topología, la toma de fotografías aéreas con fines

cartográficos o corrección de imágenes transmitidas por redes que contemplan pérdida de datos [14] [15].

5.2 Delimitación

Una vez completada la nube de puntos críticos final, el segundo paso es realizar una delimitación de cada uno de los átomos individualmente.

Esta delimitación empieza recorriendo toda la nube de puntos y haciendo una división de ésta en conjuntos más pequeños. Después, separa los conjuntos que no estén conectados entre ellos, es decir, se separan cada uno de los subconjuntos que no tienen ninguna conexión. Una conexión se forma cuando uno o más puntos críticos son consecutivos en cualquier dirección.

Una vez separada la nube de puntos principal en los distintos subconjuntos se estudia cada una de ellos por separado del siguiente modo.

Se recorre cada subconjunto eliminando los puntos críticos por cercanía que no tengan un índice de vecindad de 8 (índice de vecindad máximo), esto es, aquellos que no estén rodeados completamente por puntos críticos originales. Se considera que estos puntos con índice de vecindad máximo se deben a errores en la medición original o en el algoritmo de análisis. Una vez eliminados, se recorre cada uno de los puntos de este nuevo subconjunto, detectando y separando de nuevo en subconjuntos más reducidos, siguiendo la misma técnica utilizada por el paso anterior (aclimatación, apartado 5.1). Se trata de un algoritmo iterativo obteniendo en cada paso un subconjunto de puntos más pequeño y definido.

Finalmente, cuando ya se han obtenido los subconjuntos finales, éstos se ajustan a un rectángulo mínimo que contenga todos los puntos de cada subconjunto de puntos críticos. Para ello, se realiza este ajuste mediante el análisis de los subconjuntos, buscando en cada uno de los perfiles que los contienen y obteniendo así una nueva matriz por cada subconjunto que se convertirá en la matriz de datos de cada átomo de manera individual.

5.3 Modelado

El modelado se lleva a cabo una vez que se dispone de la matriz de datos de cada uno de los átomos obtenida en el apartado anterior, después, se procesan estas matrices siendo así posible modelar cada átomo individual.

Para ello, se almacena la posición exacta del punto central del átomo en la matriz global junto con la base y la altura del rectángulo que conforma su matriz. Cabe destacar que la matriz de datos de cada átomo no sólo contiene los puntos críticos de los mismos, sino también todos los puntos de la matriz global delimitados por este rectángulo. Cada uno de los átomos tendrá un identificador único, constituido por el nombre del fichero donde se obtuvo y los valores x e y de su centro.

A partir de esta matriz de datos se obtienen ciertas características que se han considerado relevantes para la caracterización de cada átomo individual. Esto es, se

obtienen los puntos máximos y mínimos de la matriz del átomo, así como sus perfiles siguiendo la misma metodología utilizada para la matriz global. Es decir, se obtienen los perfiles individuales de la matriz de datos del átomo y se analizan individualmente para así obtener otros datos relevantes como son la velocidad de crecimiento y de decrecimiento de cada uno de los perfiles, la media y la mediana de los puntos de la matriz y de cada perfil, la altura (valor z de los puntos) en los que se producen cambios de convexidad, la distancia entre el punto máximo y mínimo o la ganancia máxima y mínima (recordamos que el concepto de ganancia fue explicado en el apartado 4.4.5).

Todos estos datos serán almacenados en la clase que modela el átomo ya que serán necesarios para su posterior clasificación.

5.4 Clasificación

Tras el modelado de cada uno de los átomos y poder hacer la clasificación, es necesario proporcionar al sistema el elemento que representa cada uno de estos átomos analizados. En las imágenes en las que se basa este proyecto, existen tres tipos de átomos: carbono (C), hidrogeno (H) y oxígeno (O), exceptuando una serie de imágenes que también contienen una molécula de agua (H_2O).

Una vez identificados y etiquetados cada uno de los átomos, se separa el conjunto de datos para entrenamiento y test. El conjunto de entrenamiento se compone de un porcentaje configurable de los datos y el resto será utilizado para el proceso de test del modelo de red neuronal.

El sistema de clasificación implementado recibe estos datos con ciertas propiedades y, por ello, se genera un archivo que contendrá los parámetros y características necesarias para la clasificación. Se creará un fichero en formato *csv* que contiene en cada línea la modelización de un átomo, en primer lugar, el identificador único del átomo, seguido de cada uno de los parámetros seleccionados para la identificación del mismo y, por último, la clase o el elemento al que corresponde. Como se puede ver en el siguiente esquema:

```
Id;característica 1;característica 2;característica 3;.. ;característica n;class
Id;característica 1;característica 2;característica 3;.. ;característica n;class
Id;característica 1;característica 2;característica 3;.. ;característica n;class
....
n Átomos
```

Como se ha explicado previamente en el apartado 2.5, para la clasificación de los distintos átomos se ha desarrollado una red neuronal, se ha seleccionado un perceptrón multicapa con *backproagation*, ya que se trata de una red arto probada y verificada.

Además, tras diversas pruebas estudiadas, se ha optado por utilizar un MLP (*Multi Layer Perceptron*) con una única capa intermedia y un número de entradas igual a los datos obtenidos del modelado de átomos.

Cabe destacar que algunas de las imágenes del conjunto de datos presentaban cierta anomalía, en concreto, algunas de ellas se midieron cuando la molécula de agua (H_2O) se adhirió a la punta del microscopio y algunos átomos de carbono presentaban propiedades

muy distintas a las de los átomos medidos sin este tipo de anomalía, lo que confunde a la red neuronal y produce una alta tasa de error en el test final, como se verá en el apartado 6.3 de pruebas y resultados.

5.5 Diseño e implementación

Al igual que en el apartado de análisis, se describen los detalles del diseño e implementación de la aplicación desarrollada para la delimitación, modelado y clasificación de los átomos individuales extraídos de las imágenes, proceso detallado anteriormente. En concreto, se presenta el diagrama UML y se describe la funcionalidad de cada una de las clases así como las relaciones entre las mismas y se muestran los casos de uso básicos de la aplicación.

En este caso, también se ha seleccionado el lenguaje de programación Java sobre su versión 1.7, por ser multiplataforma y permitir ser utilizado en la mayoría de dispositivos, que junto con un sistema de representación intuitivo y cómodo, facilita al usuario las herramientas necesarias para este tipo de clasificación.

Al igual que la aplicación anterior se ha seguido un desarrollo incremental e iterativo que permite obtener un software funcional en cada una de estas iteraciones, agregando nuevas funcionalidades y características en cada una de las posteriores entregas.

Cabe destacar que las configuraciones referentes a la red neuronal, como son el número de capas ocultas o la constante de aprendizaje, las tolerancias de los diferentes algoritmos o las características seleccionadas para el modelado de los átomos, se obtienen de un fichero de configuración en el tiempo de ejecución, permitiendo así modificar estas configuraciones y realizar diversas pruebas.

5.5.1 Diagrama UML

En la Figura 8 se muestra el modelo UML que siguen las clases más características de la aplicación *AtomClassifier* implementada para el modelado y la clasificación de átomos.

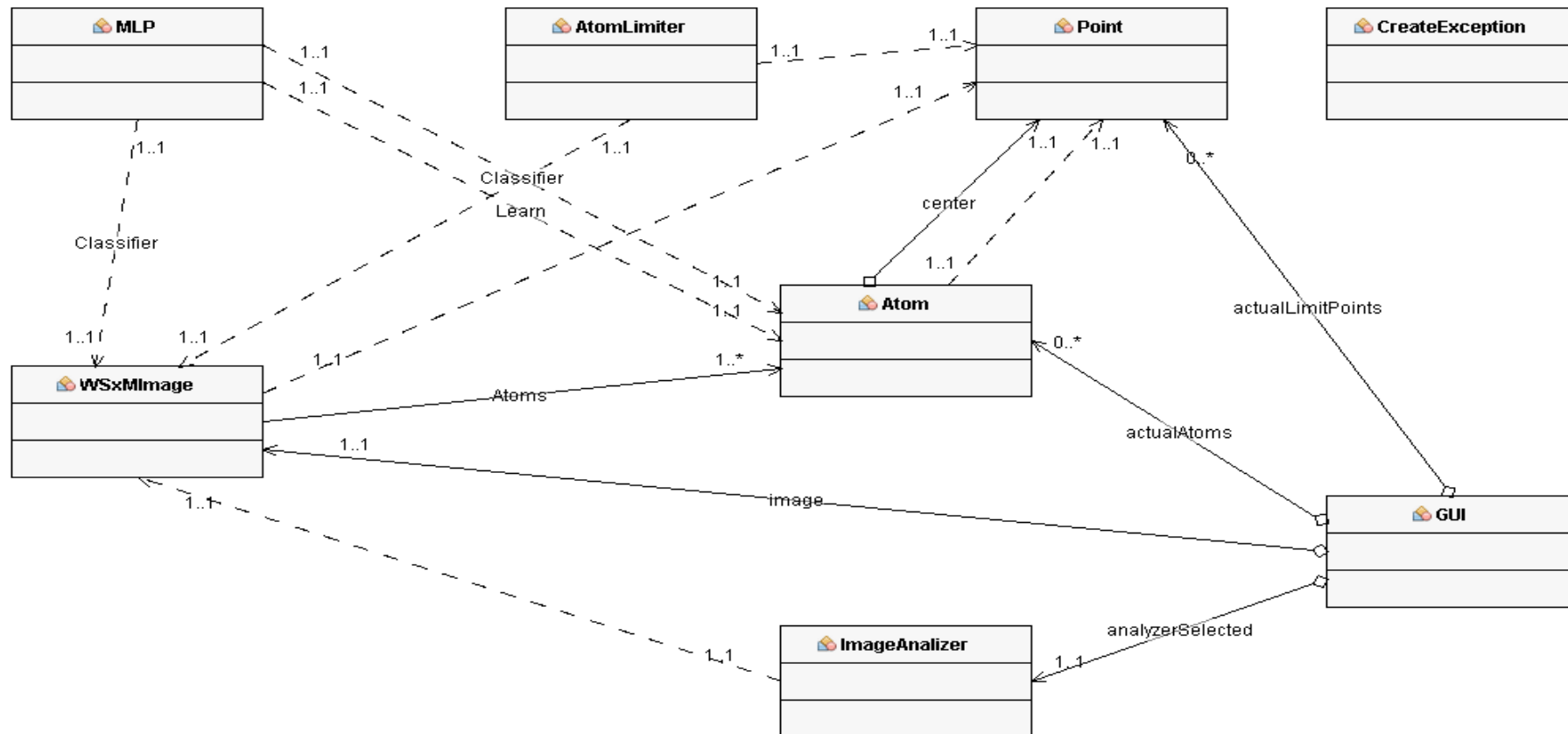


Figura 8. Diagrama UML de la aplicación AtomClassifier.

5.5.2 Diseño de clases

En este apartado se detallan las diferentes clases desarrolladas de la aplicación *AtomClassifier*. Esta aplicación se ha diseñado en base al sistema de análisis de puntos críticos final, ya que es el método con el que se han obtenido mejores resultados, como se verá en la sección 6.

Por ello, se ha reutilizado de la aplicación *AtomDetector*, anteriormente explicada, la clase *WSxMImage*, la cual contiene la matriz de datos completa junto con los datos recopilados de ella. También, se ha utilizado de ésta el analizador final, pero eliminando la interfaz *Analyzer*, y en su lugar se ha implementado la clase estática *ImageAnalyzer* que implementa el algoritmo del analizador.

Siguiendo la misma metodología de organización de la aplicación *AtomDetector* (apartado 4.5.2) se han creado una serie de paquetes bien definidos que engloban todas las partes del software, los cuales son los siguientes.

- El primero, el paquete *data* que contiene las clases que manejan los datos obtenidos, en él se incluyen las clases *WSxMImage*, *Atom* y *Point*.
- Un paquete específico para las excepciones lanzadas por los métodos de creación de imágenes, que contiene la clase *CreateException*.
- Para almacenar las clases relacionadas con la delimitación de los átomos individualmente se utiliza el paquete *detection*. En este paquete también se incluye, junto a la clase *AtomLimiter* que contiene los métodos de delimitación, la clase *ImageAnalyzer* que como se ha explicado con anterioridad implementa el modelo de análisis detallado en el apartado 4.4.5 de este documento.
- El paquete *classifier* que contiene las clases relacionadas con la red neuronal implementada.
- Por último, al igual que en la aplicación *AtomDetector*, se incluye el paquete *ui* que contiene las clases que modelan la interfaz de usuario.

Una vez detallada la organización de las clases, se procede a realizar una breve descripción de cada una de éstas. En este caso, puesto que algunas clases ya han sido estudiadas anteriormente, no se detallan en este apartado. En concreto, las clases *WSxMImage*, *ImageAnalyzer* y *Point*, que se estudiaron en los apartados 4.5.2.1, 4.5.2.4 y 4.5.2.2, respectivamente.

5.5.2.1 Atom

Esta clase modela un átomo individual, la cual contiene la matriz de datos, además de sus características y propiedades. Posee métodos de representación gráfica y sobrescribe el método *toString* que poseen todos los objetos Java, para que así genere una entrada en el fichero de datos que más tarde utilizará el modelo de aprendizaje implementado.

El creador de esta clase implementa los procesos explicados en el apartado 5.3 del modelado y clasificación. Para su creación es necesaria la matriz de datos obtenida mediante *AtomLimiter*.

5.5.2.2 AtomLimiter

AtomLimiter es la clase responsable de la delimitación de cada uno de los átomos de manera individual, partiendo siempre de la nube de puntos generada mediante la clase *ImageAnalyzer*. Esta clase implementa el algoritmo de delimitación explicado en la sección 5.2.

5.5.2.3 CreateException

A la hora de leer algún archivo erróneo o que no existe se lanza la excepción *CreateException*. Ésta puede ser lanzada por los métodos de creación de la clase *Atom* y *WSxMImage*.

5.5.2.4 MLP

Es la clase que implementa la red neuronal, que como ya se ha explicado con anterioridad, se ha utilizado un perceptrón multicapa con *backpropagation*, de ahí proviene el nombre de MLP: *Multi Layer Perceptron*. Esta clase posee los métodos de aprendizaje y explotación junto con una función de guardado, que permite exportar una red neuronal que ha pasado la fase de entrenamiento y así que pueda ser utilizada posteriormente.

5.5.2.5 GUI

Es la interfaz gráfica de la aplicación. Ésta es la clase ejecutable que genera la interfaz de usuario y otorga una visión de cada uno de los análisis, imágenes cargadas y resultados de la clasificación.

En la siguiente sección (apartado 5.5.3), se explican en detalle los casos de uso básicos para el correcto funcionamiento de la misma.

5.5.3 Casos de uso

Toda la funcionalidad descrita en el apartado anterior se ha dotado también de una interfaz gráfica de fácil uso para el usuario, la cual permite analizar, delimitar y clasificar átomos individualmente analizando las imágenes generadas por la aplicación *WSxM*. Además, esta aplicación posee un gran número de opciones y configuración, por lo que en este apartado se detallan todos los casos de uso de la misma, incluyendo también ejemplos de uso de la interfaz para su correcta utilización.

A continuación en la Figura 9 se muestra el diagrama de casos de uso de la aplicación, y se explicará en detalle cada uno de ellos.

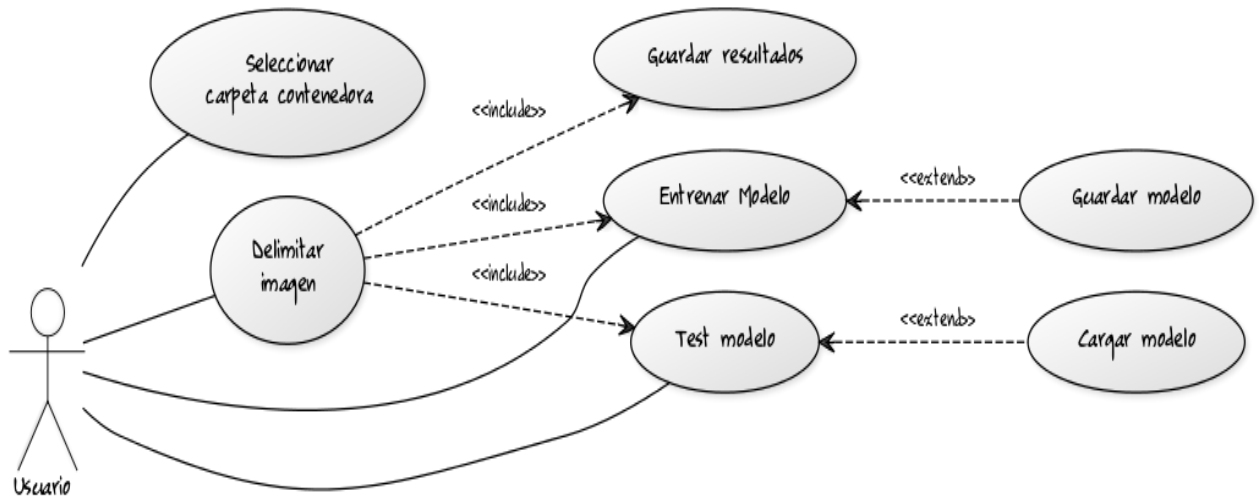


Figura 9. Diagrama de casos de uso AtomClassifier.

Al igual que en el apartado 4.5.3, primero, se describirán los puntos de interacción del usuario en la interfaz (marcados en la Figura 10), para así poder tener una referencia sobre la que documentar los distintos usos del software.

Select Folder (1)

Este botón se utiliza para seleccionar la carpeta que contiene las imágenes seleccionadas para almacenar, entrenar o evaluar, según desee el usuario.

Load Model (2)

Mediante este botón se carga un modelo ya entrenado previamente, éste se encuentra almacenado en un fichero de texto generado por esta misma aplicación, en la opción de *Save Model*.

Save Model (3)

Tal y como se ha explicado, este botón permite guardar un modelo ya entrenado en un fichero de texto, para su posterior evaluación.

Original Image (4)

De manera análoga que en la aplicación *AtomDetector*, es una pequeña ventana dentro de la interfaz donde se muestra la representación gráfica de la imagen a analizar.

Result Image (5)

En este apartado de la interfaz se representa la imagen original destacando los átomos identificados o la clasificación de los mismos.

Statistics (6)

Se trata de un cuadro de texto donde se mostrarán los datos estadísticos de la imagen cargada.

Analyze (7)

Para iniciar con el análisis, delimitación y modelado de los átomos de la imagen actual se utiliza este botón.

Save to File (8)

Con este botón se almacenan los átomos delimitados y modelados en el último análisis en un fichero.

Next Image (9)

Este botón permite al usuario seleccionar la siguiente imagen dentro del directorio especificado.

Train (10)

Es posible entrenar la red neuronal utilizando el botón *Train*. Para ello, la aplicación utiliza como instancias de entrenamiento un fichero de datos seleccionado.

Evaluate (11)

Este botón permite seleccionar el fichero que será posteriormente evaluado mediante el modelo actual.

Log (12)

Al igual que en la aplicación anterior, en este cuadro de texto se muestran los *logs* de la aplicación.

Exit (13)

De nuevo, para salir de la aplicación, existe el botón *Exit*, que libera todos los recursos utilizados y la cierra.

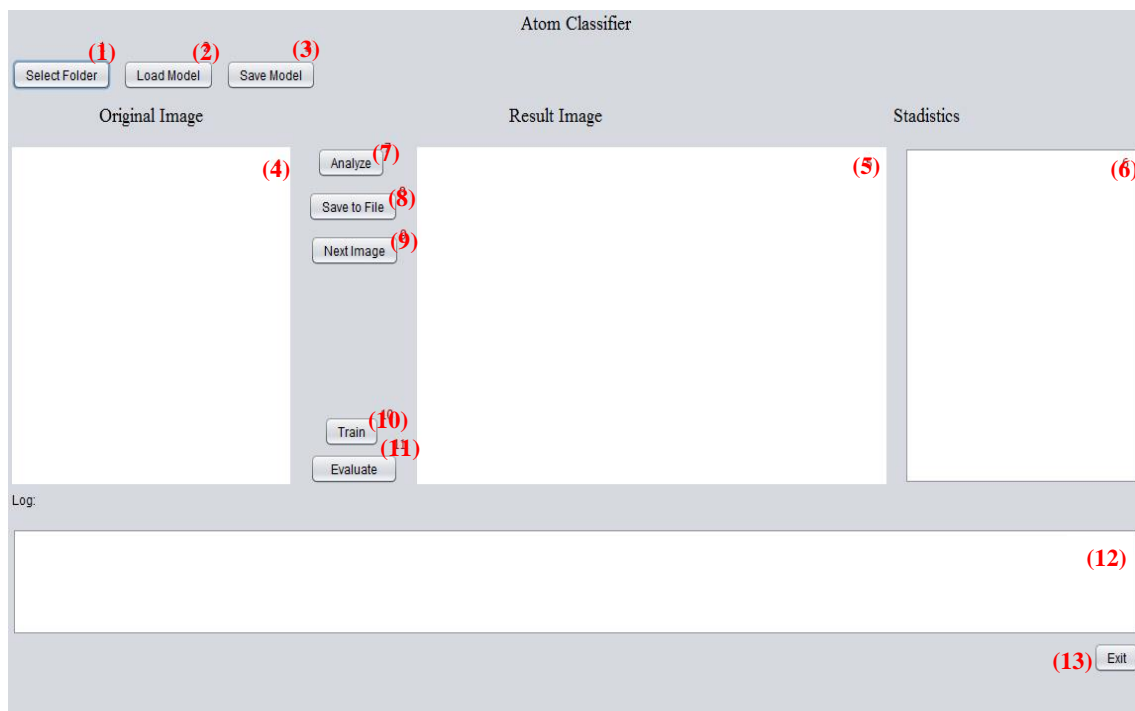


Figura 10. Ventana principal de la aplicación AtomClassifier.

5.5.3.1 Selección de la carpeta contenedora y navegación

Para comenzar el proceso de análisis o de clasificación es necesario seleccionar una carpeta donde se encuentran las distintas imágenes utilizadas en la aplicación. Una vez seleccionada la carpeta contenedora, la aplicación recorrerá las distintas imágenes de la misma tanto para analizarlas como para, posteriormente, clasificar los distintos átomos.

De manera que, pulsando el botón de seleccionar directorio (*Select Folder*, punto (1) en la Figura 10), se creará un diálogo adicional en el que el usuario podrá seleccionar la carpeta que contiene las imágenes que desea analizar o clasificar.

El usuario podrá navegar entre las distintas imágenes pulsando el botón *Next Image* (punto (9) en la Figura 10). Además, cada una de estas imágenes podrá ser analizada y de éstas el usuario podrá almacenar los átomos analizados mediante el botón *Save to File* (punto (8) en la Figura 10), que almacenará cada átomo identificado en la imagen en un fichero de salida. Dicho fichero será seleccionado o creado si es la primera vez que se pulsa en la sesión.

5.5.3.2 Delimitación de imágenes

Una vez que el usuario ha seleccionado una carpeta y una imagen dentro de ella, el siguiente paso del análisis y clasificación es la delimitación y modelado de cada uno de los átomos detectados.

Para ello, el usuario, después de seleccionar una imagen mediante el procedimiento anterior, deberá pulsar el botón *Analyze* (punto (7) en la Figura 10) y el propio software se encargará de analizar la imagen, detectar los puntos críticos, delimitar y modelar cada uno de los átomos identificados en la imagen. Tras este análisis de la imagen, se muestran en el campo *Result Image* correspondiente (punto (5) en la Figura 10)) el resultado junto con las fronteras de los átomos detectados.

El usuario puede realizar varias pruebas y una vez que considere valiosa la información recogida por la aplicación, podrá almacenar los átomos detectados mediante el botón *Save to File* (punto (8) en la Figura 10). Si es la primera imagen que se desea guardar en la sesión, el programa mostrará un diálogo adicional para seleccionar dónde desea guardar el fichero así como su nombre.

En la Figura 11 se muestra el resultado de este análisis de una imagen concreta.

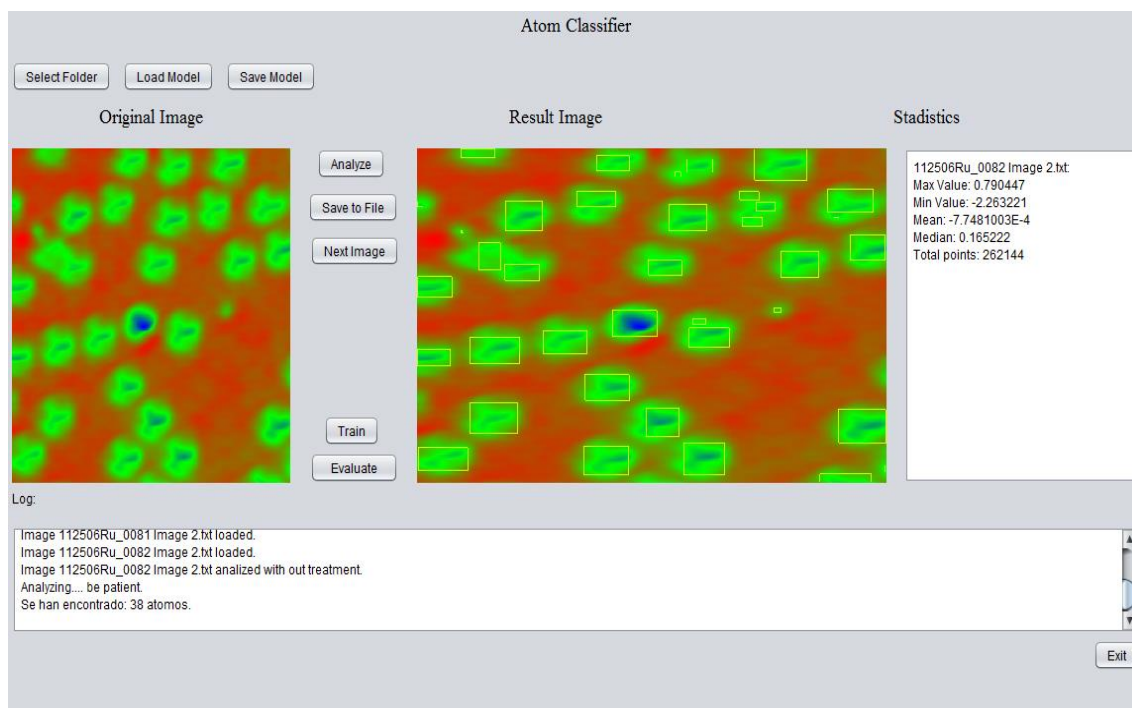


Figura 11. Delimitación de una imagen mediante la aplicación AtomClassifier.

En la Figura 11 se muestra un ejemplo en el que visualmente la aplicación detecta la mayoría de los átomos, mostrando en el cuadro de *Log* que se han detectado un total de 38 átomos.

5.5.3.3 Procesamiento de imágenes y generación de ficheros

Mediante el proceso anterior se ha demostrado cómo es posible analizar una imagen concreta y delimitar los átomos de la misma. Una vez analizada, esta aplicación permite almacenar los átomos detectados en un fichero de datos. Para realizar este proceso bastará con pulsar el botón *Save to File* (punto (8) en la Figura 10), mediante el cual se pregunta al usuario en qué fichero desea almacenar los datos. El formato de este fichero será como el descrito en el apartado 5.4 del presente documento.

Tras detectar y analizar los átomos de una imagen, el usuario podrá pulsar el botón *Next Image* (punto (9) en la Figura 10) y analizar la siguiente imagen de la carpeta, pudiendo así almacenar distintos átomos y generar un modelo más adelante.

El único proceso no automatizado es el de identificar el elemento de cada uno de los átomos para que la red neuronal los tome como referencia tanto para el entrenamiento como para el cálculo del error en la fase de test. Para llevar a cabo esto, es necesario que el usuario escriba en el fichero de datos generado la clase de cada uno de ellos. En el apartado de mejoras (apartado 7.2) se expone una posible solución a este inconveniente.

5.5.3.4 Entrenamiento y almacenaje de modelo

El siguiente paso será realizar un correcto entrenamiento de la red neuronal y almacenar el modelo entrenado por si fuese necesario utilizarlo en otro momento.

El modelo se guardará haciendo uso de la interfaz *Serializable* de Java, la cual provee los métodos necesarios para almacenar un objeto completo en un archivo binario.

Una vez almacenadas las imágenes que se desean utilizar para la fase de entrenamiento, basta con que el usuario pulse el botón *Train* (punto (10) en la Figura 10). En ese caso, el programa cerrará el fichero de datos donde estén guardados los átomos analizados y utilizará este mismo fichero para entrenar la red neuronal. Las configuraciones de esta red neuronal, tal y como se ha explicado anteriormente, están almacenadas en un fichero de configuración que es leído en tiempo de ejecución, por lo que si se necesita realizar algún ajuste sobre estos parámetros basta con modificar este archivo antes de comenzar el entrenamiento.

Al finalizar la fase de entrenamiento, el programa lo indicará mediante un mensaje en el apartado de *Log*. El modelo será almacenado en memoria, en el propio objeto que implementa la red neuronal, de manera que si el usuario desea guardarlo debe pulsar el botón *Save Model* (punto (3) en la Figura 10); en caso contrario, el modelo se descartará al cerrar la aplicación.

5.5.3.5 Carga y test del modelo

Una vez entrenada la red neuronal, podrá ser utilizada para clasificar átomos de una imagen cargada. Tanto si el modelo es cargado mediante el botón *Load Model* (punto (2) en la Figura 10) como si el modelo ha sido entrenado en la misma sesión, es necesario que esté presente en la aplicación.

Para tratar de clasificar una imagen es necesario seleccionar una imagen, primero, cargando una carpeta contenedora y, a continuación, elegir la imagen mediante el botón *Next Image* (punto (9) en la Figura 10). Tras esto, es necesario pulsar el botón *Analyze* (punto (7) en la Figura 10) para detectar los átomos de las imágenes y, una vez detectados y teniendo el modelo de red neuronal cargado, basta con pulsar el botón *Evaluate* (punto (11) en la Figura 10) para realizar una clasificación de los átomos de la imagen seleccionada. A continuación, en la Figura 12, se muestra un ejemplo de esto.

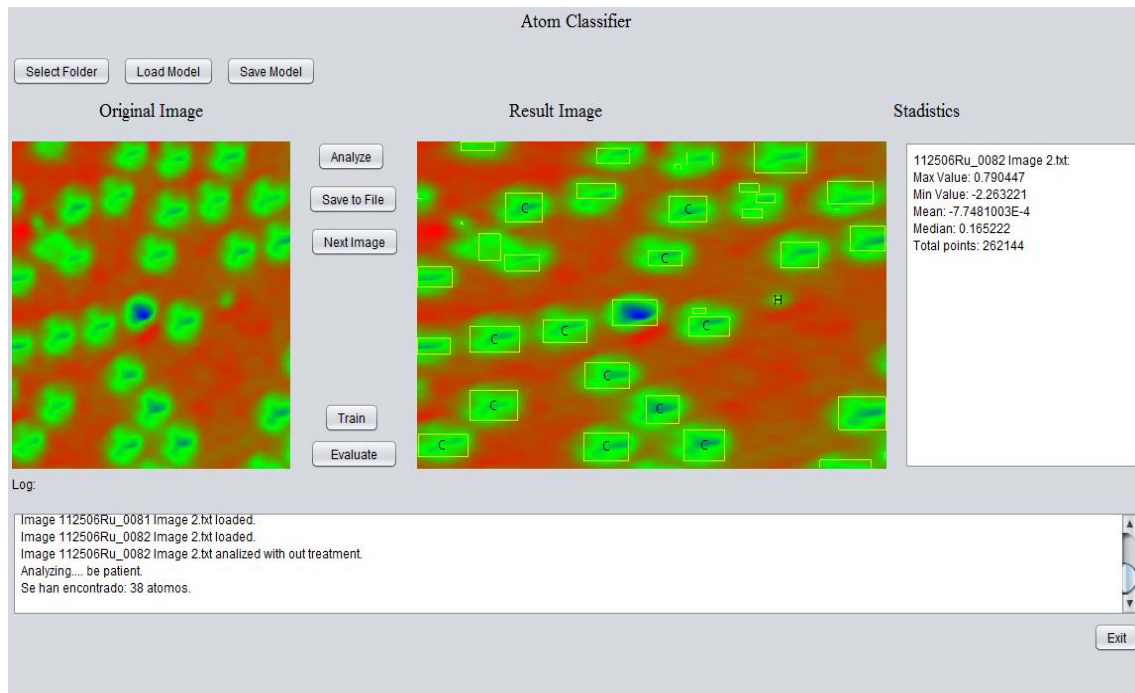


Figura 12. Clasificación de una imagen mediante la aplicación AtomClassifier.

Como se puede observar en la Figura 12, el programa clasifica los átomos colocando el tipo de elemento según sea carbono (C), hidrógeno (H), oxígeno (O) o agua (H₂O) sobre el centro del átomo detectado. Los átomos detectados a los que no se les asigna ningún valor son aquellos en los que la red neuronal no tiene una solución viable. Como se detalla más adelante en el apartado 6, no se consiguen resultados muy satisfactorios ya que como se puede observar en este ejemplo el algoritmo ha clasificado 19 átomos de los 38 posibles.

6. Pruebas y resultados

En este apartado se explican los experimentos y pruebas más representativas realizadas sobre la implementación explicada en los capítulos anteriores.

En primer lugar, se muestra uno de los problemas que más esfuerzo ha llevado solventar en este trabajo. Se trata de un problema bastante común a la hora de la detección automática de átomos en este tipo de imágenes, el cual se ha sido referenciado en este trabajo como imagen escalonada. Es decir, en una imagen que contiene en un primer nivel una serie de átomos, mientras que a un segundo nivel, más alto o más bajo, se encuentra otro conjunto diferente, los algoritmos tradicionales como el algoritmo de planos paralelos, no es capaz de distinguir estas dos alturas produciendo una detección poco precisa.

En segundo lugar, se ha entrenado una red neuronal sólo con imágenes en las que la punta del microscopio tiene una molécula de agua (H_2O) adherida, obviando el resto.

Por último, se llevará a cabo una prueba completa, seleccionado al azar el 50% de las imágenes del conjunto completo y realizando el test sobre una imagen que no tiene adherida a la punta del microscopio la molécula de H_2O .

6.1 Detección de puntos críticos en imágenes escalonadas

En este apartado se muestra el resultado de aplicar distintos métodos de análisis de puntos críticos sobre una imagen escalonada, la cual se caracteriza por presentar dos alturas de átomos diferentes. Puesto que en el conjunto de datos original no se dispone de imágenes con estas características, se ha seleccionado una de ellas al azar y se ha modificado su matriz de datos para forzar este escalón que la caracteriza. Para llevar a cabo esta simulación, se han restado 10 puntos a la mitad derecha de la matriz obteniendo el resultado de la Figura 13 y Figura 14, en el que se emplea el modelo de máximos y mínimos (*MinMax*) frente al modelo final completo (explicado en el apartado 4.4.5), respectivamente.

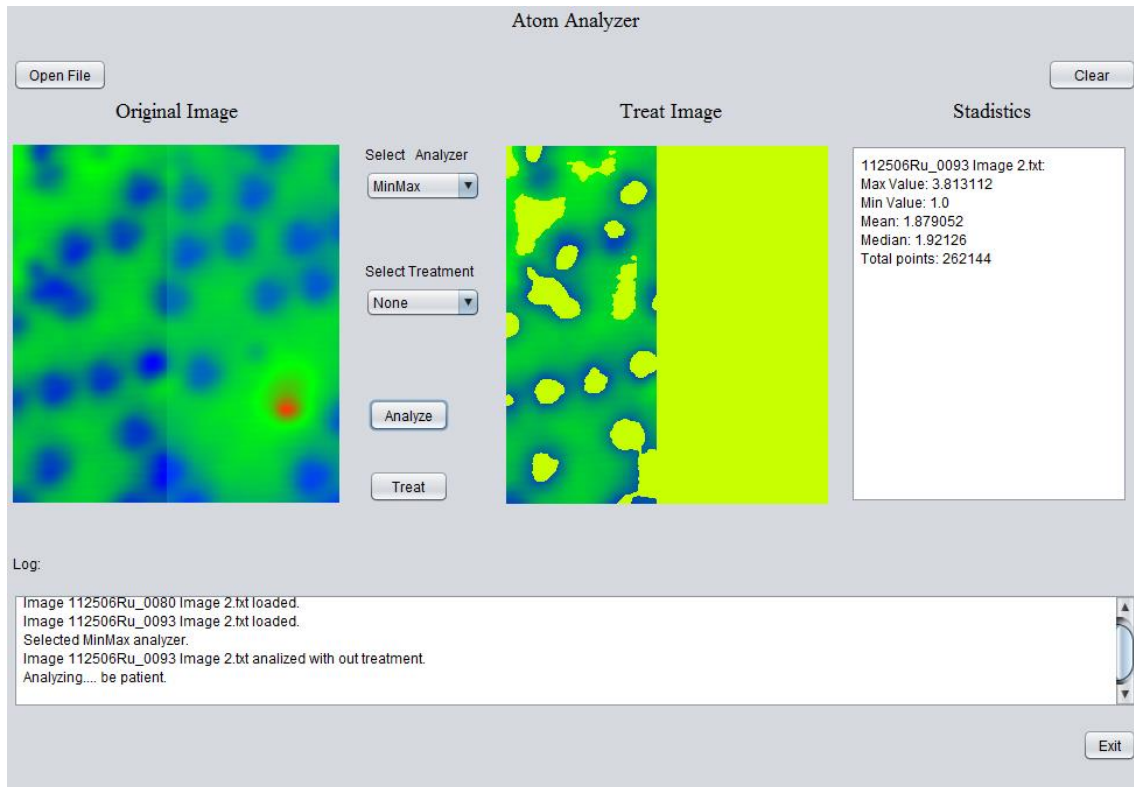


Figura 13. Detección de puntos críticos de una imagen escalonada utilizando MinMax.

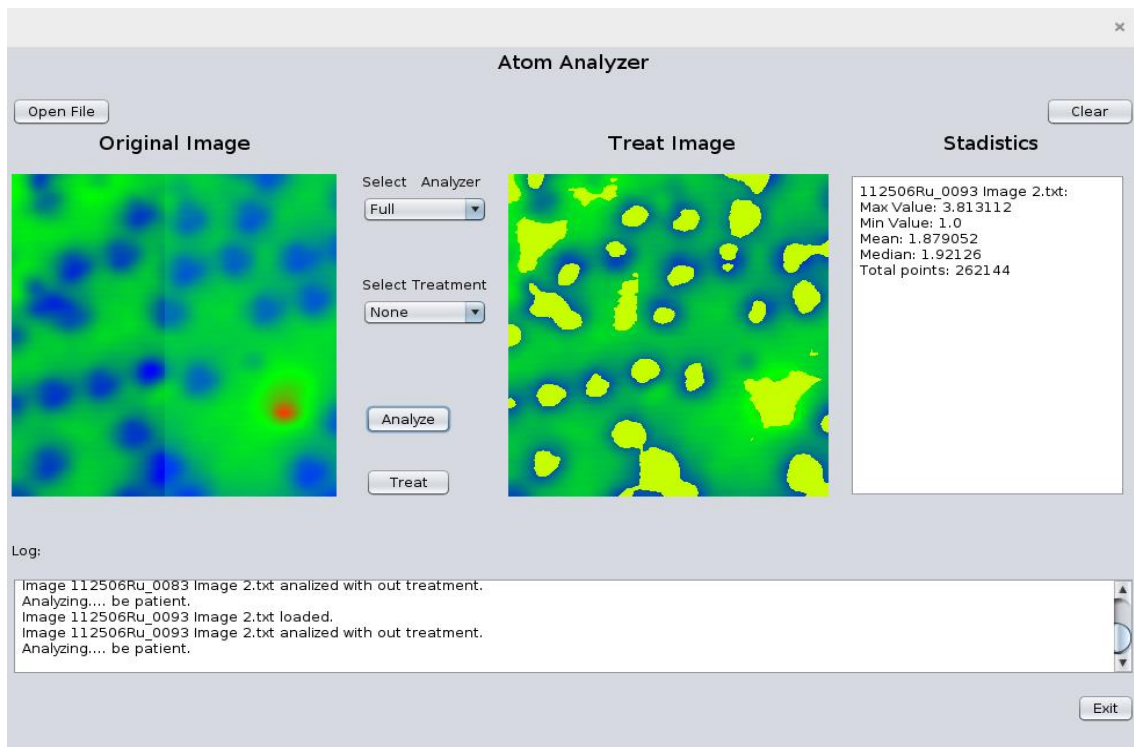


Figura 14. Detección de puntos críticos de una imagen escalonada utilizando el modelo final.

Como se puede observar en la Figura 13 el modelo de *MinMax* no consigue diferenciar correctamente las diferentes alturas, detectando como átomos toda la parte de la derecha de la matriz que se encuentra a una altura inferior. Por el contrario, en la Figura

14, el algoritmo final consigue superar la prueba exitosamente distinguiendo los átomos de las dos alturas.

6.2 Entrenamiento y test sobre imágenes del mismo conjunto

En el experimento desarrollado en este apartado, se ha utilizado únicamente un conjunto de imágenes en las cuales la punta del microscopio tenía adherida una molécula de agua durante las mediciones. De manera que el conjunto de imágenes para el entrenamiento y el test son similares, esperando así minimizar los problemas de esta anomalía sobre los resultados. El conjunto de imágenes resultante se compone de un total de 12 imágenes distintas, con 468 átomos a analizar, de las cuales 11 se han utilizado para la fase de entrenamiento dejando la imagen restante para el test.

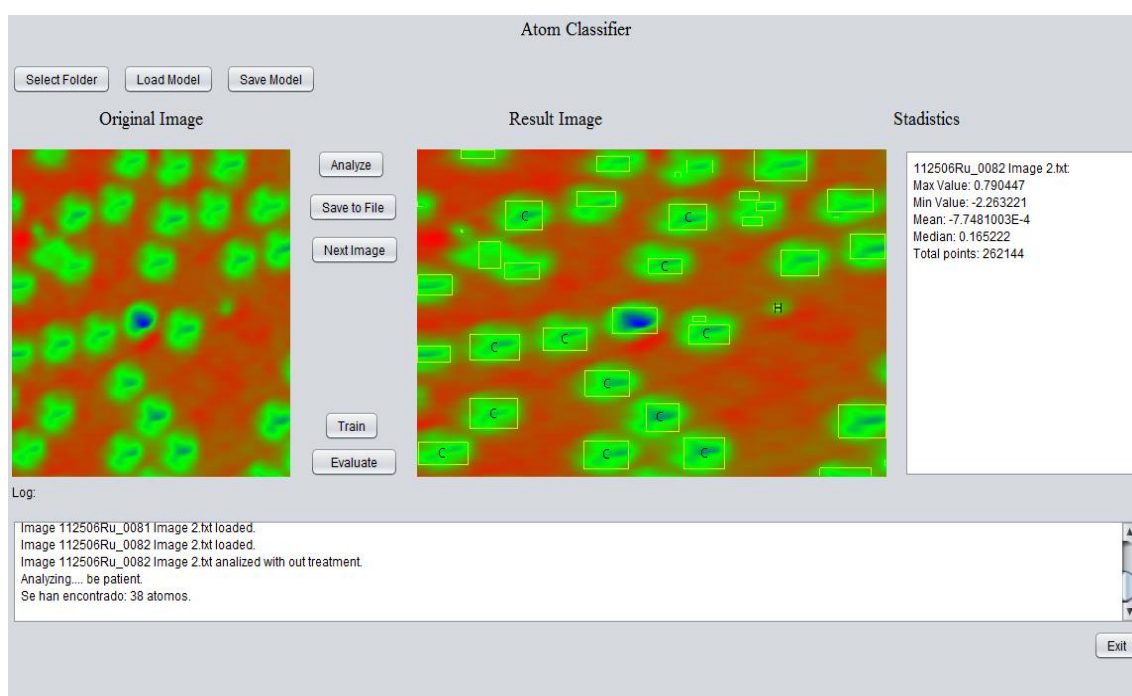


Figura 15. Entrenamiento y test de una imagen perteneciente al conjunto medido con H_2O adherido.

En este caso de prueba se obtiene la Figura 15, donde se puede observar que la red neuronal detecta correctamente la mayoría de los átomos de carbono (añadiendo una C dentro de su rectángulo) e hidrógeno (añadiendo una H), mientras que los átomos de oxígeno (O) y algunos de carbono no son reconocidos. En este caso se obtiene un error del 51,28%.

Después de un estudio de las posibles causas de este error, podemos llegar a la conclusión de que el método de delimitación está muy ligado a la frontera de las nubes de puntos críticos, por lo que existen ciertos átomos que a pesar de que pertenezcan al mismo elemento, el sistema de delimitación les otorga tamaños distintos.

6.3 Entrenamiento y test sobre imágenes sin restricciones

En este apartado se realiza una prueba utilizando el conjunto de datos completo de imágenes, teniendo en cuenta tanto las imágenes con la punta del microscopio limpia como las que tenían adherida una molécula de agua en la medición. En este caso, para la fase de entrenamiento se han seleccionado el 50% de las imágenes dejando el resto para el test.

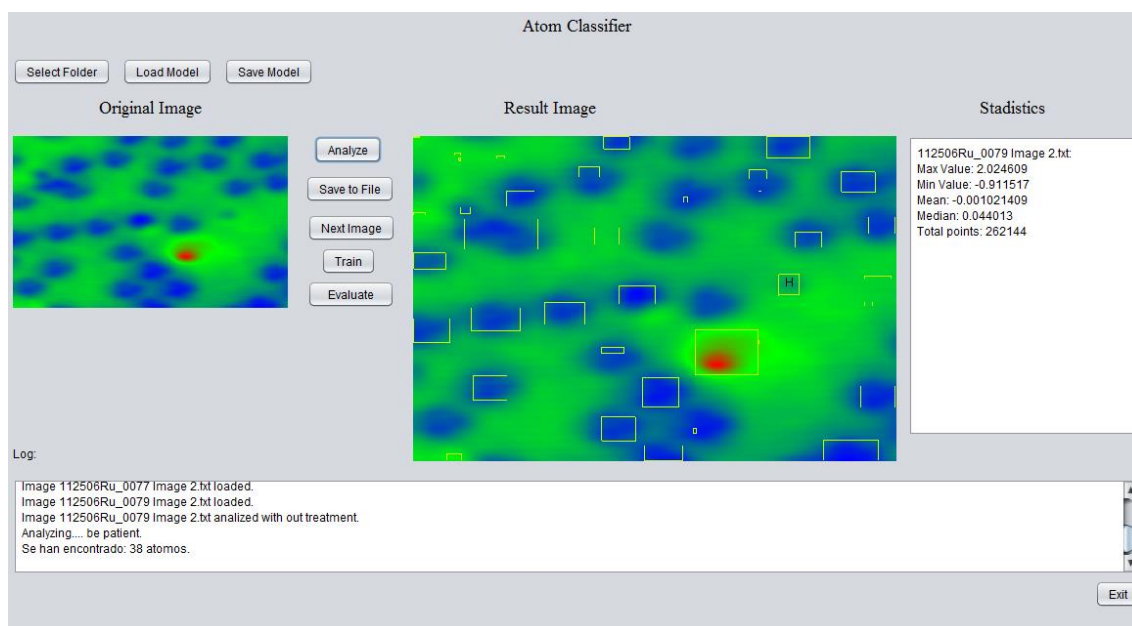


Figura 16. Entrenamiento y test de una imagen sin restricciones.

Como se puede observar en la Figura 16, la red neuronal solamente es capaz de reconocer el átomo de hidrógeno (pues aparece una H en el rectángulo). Esto se debe a que átomos muy distintos son del mismo elemento provocado por las mediciones tan dispares que realiza el microscopio respecto al estado de su punta, lo que hace que la red neuronal no sepa realizar una clasificación acertada.

7. Conclusiones y trabajo futuro

El objetivo de este trabajo es el diseño de un conjunto de algoritmos y un software capaz de diferenciar entre diferentes tipos de átomos de elementos distintos, así como la delimitación y caracterización de cada átomo individualmente.

El trabajo desarrollado abarca ciertos conceptos teóricos necesarios para comprender el proyecto, métodos de medición en escalas nanométricas, técnicas de tratamiento y análisis de imágenes, así como ciertos principios para el aprendizaje automático. Además de aspectos para el análisis, modelado y clasificación, la implementación de software para la ejecución de simulaciones sobre diferentes tipos de imágenes y configuraciones, la visualización de las mismas y el análisis estadístico.

Recapitulamos a continuación los aspectos más relevantes del trabajo realizado, y planteamos las múltiples líneas de trabajo futuro que se abren a partir del trabajo desarrollado hasta aquí.

7.1 Conclusiones

Durante el desarrollo de este proyecto se han alcanzado con éxito gran parte de los objetivos propuestos, se ha conseguido una detección de átomos completa, de forma óptima y con mejores resultados que el software comercial, solventando problemas que no podían ser resueltos de forma clara en este tipo de aplicaciones, como son las imágenes escalonadas o con mucho ruido.

Por otra parte se ha desarrollado un software completo, con una interfaz amigable y de fácil uso para un usuario sin conocimientos en este campo, facilitando así las tareas de análisis y clasificación de imágenes de microscopía a escala nanométrica. Además, se ha conseguido la clasificación de distintos átomos en las imágenes con una precisión aceptable.

En la elaboración de este trabajo, no sólo se ha aprendido a aplicar conceptos de análisis, modelado y clasificación automática para la resolución de problemas propios de la ingeniería, sino que también se han adquirido nuevos conocimientos para la extracción automática de información, el tratamiento de imágenes en escalas nanométricas y el manejo de grandes volúmenes de datos. Se han empleado nuevos programas y bibliotecas que hasta ahora no se habían utilizado, se han adquirido conocimientos sobre el campo de la Nanotecnología y muchas otras competencias que, sin duda, serán útiles a lo largo de la carrera profesional.

7.2 Trabajo futuro

A lo largo del trabajo se han ido citando varias posibilidades para un posible trabajo futuro. En este apartado se exponen de forma sintetizada todas ellas y algunas más que no se han mencionado explícitamente hasta ahora.

Para mejorar la detección de átomos, mediante el algoritmo final se consiguen muy buenos resultados, se podría implementar algún algoritmo de reconocimiento de imágenes o un mapa autoorganizado, con el que se podrían obtener mejores resultados, el mapa autoorganizado podría también usarse para mejorar la delimitación de átomos individuales obteniendo así fronteras más precisas.

Una de las mejoras pensadas para la interfaz sería el uso de un sistema dinámico para aportar la clase de los átomos de forma interactiva, y así proporcionar al usuario una nueva interfaz para seleccionar el elemento de cada átomo al mismo tiempo que se analizan las imágenes.

En la parte de clasificación podrían incluirse más algoritmos de *machine learning* como búsqueda de KNN u otros tipos de redes neuronales. De esta forma, el usuario seleccionar que algoritmo usar al comienzo del aprendizaje.

Referencias

- [1] VENTURINI, V. M.: Inteligencia Ambiental y Nanotecnología: El paso del Bit al átomo: URL: <http://www.ucasal.net/templates/unidadacademicas/ingenieria/apps/4-p62-Venturini.pdf>, 2009
- [2] PETERS, L.: Time Travel to 2029: Nanotechnology Thrives. In: *Semiconductor International* 1 (2004), Nr. 1
- [3] FANET, H.: Telecommunications: “ambient intelligence” needs nanotechnologies. In: *CLEFS CEA* 52 (2005)
- [4] I. HORCAS et al. Rev. Sci. Instrum. 78, 013705 (2007)
- [5] RUGAR, D. & HANSMA, P.: Atomic force microscopy. In: *Physics today* 43 (1990), Nr. 10, S. 23--30
- [6] SACHA, G.; GÓMEZ-NAVARRO, C.; SÁENZ, J. & GÓMEZ-HERRERO, J.: Quantitative theory for the imaging of conducting objects in electrostatic force microscopy. In: *Applied physics letters* 89 (2006), Nr. 17, S. 173122
- [7] BINNIG, G. & ROHRER, H.: Scanning tunneling microscopy. In: *Surface science* 126 (1983), Nr. 1, S. 236--244
- [8] SUGIMOTO, Y.; POU, P.; ABE, M.; JELINEK, P.; PÉREZ, R.; MORITA, S. & CUSTANCE, O.: Chemical identification of individual surface atoms by atomic force microscopy. In: *Nature* 446 (2007), Nr. 7131, S. 64--67
- [9] SALOMONE, M.: ¿Es segura la nanotecnología? [citado Jul 2008]. Disponible en: <http://www.madrimasd.org>
- [10] MEJIAS SÁNCHEZ, Y.; CABRERA CRUZ, N.; TOLEDO FERNÁNDEZ, A. M. & DUANY MACHADO, O. J.: La nanotecnología y sus posibilidades de aplicación en el campo científico-tecnológico. In: *Revista Cubana de Salud Pública* (2009), Nr. 3, S. 0--0
- [11] POOLE, C. P. & OWENS, F. J.: Introducción a la nanotecnología: *Reverté*, 2007
- [12] NAKAGAWA, Y. & ROSENFELD, A.: Some experiments on variable thresholding, *Pattern Recognition*. 11, 1979, S. 191--204
- [13] RUSS, J. C. & WOODS, R. P.: The image processing handbook. In: *Journal of Computer Assisted Tomography* 19 (1995), Nr. 6, S. 979--981
- [14] RICHARDS, J. A. & XIUPING, J.: Remote sensing digital image analysis: *Springer*. 3, 1999
- [15] ROY, D.; DEVEREUX, B.; GRAINGER, B. & WHITE, S.: Parametric geometric correction of airborne thematic mapper imagery. In: *International Journal of Remote Sensing* 18 (1997), Nr. 9, S. 1865--1887